ResearchOnline@JCU

JAMES COOK UNIVERSITY AUSTRALIA

This file is part of the following work:

Janardhanan, Sreekanth (2012) ***Integrated multi-objective management of saltwater intrusion in coastal aquifers using coupled simulation-optimisation and monitoring feedback information.*** **PhD Thesis, James Cook University.**

Access to this file is available from:

https://doi.org/10.25903/n73f%2Dp471

Copyright © 2012 Sreekanth Janardhanan

The author has certified to JCU that they have made a reasonable effort to gain permission and acknowledge the owners of any third party copyright material included in this document. If you believe that this is not the case, please email researchonline@jcu.edu.au

ResearchOnline@JCU

This file is part of the following reference:

Janardhanan, Sreekanth (2012) *Integrated multi-objective management of saltwater intrusion in coastal aquifers using coupled simulation-optimisation and monitoring feedback information.* PhD thesis, James Cook University.

Access to this file is available from:

http://researchonline.jcu.edu.au/36942/

The author has certified to JCU that they have made a reasonable effort to gain permission and acknowledge the owner of any third party copyright material included in this document. If you believe that this is not the case, please contact ResearchOnline@jcu.edu.au and quote **http://researchonline.jcu.edu.au/36942/**

JAMES COOK UNIVERSITY AUSTRALIA

Integrated Multi-objective Management of Saltwater Intrusion in Coastal Aquifers Using Coupled Simulation-Optimisation and Monitoring Feedback Information

Thesis submitted by

Sreekanth JANARDHANAN

B Tech (Agricultural Engineering, KAU, Kerala, India)

M Tech (Civil Engineering, IIT Kanpur, India)

In January 2012

for the degree of Doctor of Philosophy

School of Engineering & Physical Sciences

James Cook University

Statement of Contribution of Others

All the methodologies, concepts and results reported in this thesis are developed and written by Sreekanth Janardhanan under the overall supervision of the principal supervisor, Dr Bithin Datta. Model conceptualisation, development and modification of programs were done by Sreekanth Janardhanan, with Dr Bithin Datta providing advice on various aspects of the work. The text of the published articles in journals, conference proceedings and the thesis was written by Sreekanth Janardhanan, with Dr Bithin Datta critically reviewing the manuscripts.

Financial support was received from CRC-CARE Pty Ltd, Mawson Lakes, South Australia. Financial support included living expenses and covered research costs by means of project funds and professional development funds. The financial support includes;

Research funding:

CRC-CARE (Living Expenses, for 3 yrs)	85,500
CRC-CARE (Project fund, for 3 yrs)	15,000
CRC-CARE (Professional development fund, for 3 yrs)	15,000
Graduate Research School, James Cook University	2,800

Assistance was received from Jewel See professional editors, Queensland, in proofreading the thesis.

Acknowledgements

I wish to express my profound sense of gratitude and respect to my principal supervisor, Dr. Bithin Datta, Senior Lecturer, School of Engineering and Physical Sciences, James Cook University, Townsville, for his invaluable guidance throughout the course of this study. The submission of this thesis has become possible because of the guidance, help and moral support obtained from Dr Datta throughout the tenure of my PhD research. I am particularly thankful to him for the encouragement he gave to pursue the ideas I had in my own ways, to accomplish this work.

My sincere thanks are due to CRC-CARE Pty Ltd, Australia, for providing the financial support for my research and living expenses throughout the tenure of my study in James Cook University. I also wish to express my gratitude to Assoc. Prof. Siva Sivakugan and other faculty, staff and colleagues in the faculty of science and engineering. In particular, I would like to thank Mr. Manish Kumar Jha, Mr. Om Prakash, Assoc. Prof. Mohan Jacob, Ms Melissa Norton, Ms Paula Rodger, Ms Shirley Johnson, Ms Barbara Pannach, Ms Nadene George and Ms Fiona Wong.

Special thanks are also due to my wife, parents, family members and friends who made my life enjoyable during the course of my stay in Townsville.

J. Sreekanth

ABSTRACT

Coastal aquifers are aquifers which are hydraulically connected to the sea. They are important sources of groundwater which are often over-exploited due to the high density of population existing near the coasts. Coastal aquifers are susceptible to seawater intrusion caused by over-exploitation or other factors like sea level rise due to climate change. Carefully planned groundwater extraction and monitoring strategies are required for the optimal and sustainable use of coastal aquifers. This study develops methodologies for multi-objective optimal groundwater extraction strategies using simulation and optimisation techniques. Two conflicting objectives of management, viz, maximising the total beneficial pumping and minimising the total barrier well pumping are considered in this work. This study also develops optimal monitoring network designs for evaluating the compliance of the implemented strategies with the prescribed ones and illustrates the sequential modification of the prescribed strategies based on the feedback information from the compliance monitoring network.

A coupled simulation-optimisation framework is proposed and developed as the basic tool for deriving optimal groundwater management strategies. A three-dimensional density-dependent flow and transport simulation model FEMWATER is used to simulate the coastal aquifer responses to groundwater extraction, in terms of the saltwater intrusion levels. A large number of such simulations is performed to generate the concentration levels resulting from different combinations of pumping from the beneficial and barrier well pumping locations. This pumping-salinity dataset is used as input-output patterns to train and test surrogate models based on modular neural networks (MNN) and genetic programming (GP). Properly trained and tested surrogate models are coupled to the multi-objective genetic algorithm. The optimisation algorithm iteratively searches for the optimal groundwater extraction strategies in a number of generations and in each step, the surrogate models are run to evaluate the salinity levels resulting from the candidate pumping strategies considered. The Pareto-optimal set of solutions is evolved after a number of such generations. It is observed from the obtained results that both surrogate modelling approaches identified similar Pareto-optimal front of solutions for the coastal aquifer management problem. However, the genetic programming based surrogate modelling approach is found to have specific advantages when used in the simulation-optimisation framework.

One of the main concerns regarding surrogate modelling based simulation-optimisation is the non-reliability issues associated with the optimal solutions resulting from the approximation involved and predictive uncertainty of the surrogate models. In this study a methodology is developed for obtaining reliable solutions to coastal aquifer management by overcoming the predictive uncertainty of the surrogate models. In this approach, an ensemble of surrogate models is developed to predict the aquifer responses to pumping. Bootstrap samples of pumping-salinity patterns are used to train and test different surrogate models using genetic programming. The number of surrogate models in the ensemble is determined by an uncertainty criterion. All the surrogate models in the ensemble are independently coupled to the multi-objective genetic algorithm, and a multiple-realisation optimisation approach is utilised to derive reliable optimal pumping strategies for coastal aquifer management. Reliability of optimal solutions is defined in terms of the percentage of the surrogate models, for which the imposed constraints are satisfied in deriving the pumping

solutions. From these results, it is observed that optimal solutions with increased levels of reliability can be obtained using this proposed approach.

The ensemble surrogate based methodology is further extended to address coastal aquifer management under parameter uncertainty. Uncertainty in the values of hydraulic conductivity and annual aquifer recharge are considered. The realisations of hydraulic conductivity and aquifer recharge are sampled from their respective distributions using Latin hypercube sampling. Bootstrap samples of pumping-salinity patterns generated using the numerical simulation model over different realisations of the uncertain parameters are used to train and test different surrogate models in the ensemble. Thus, surrogate models in the ensemble have different predictive capabilities in different regions of the parameter-decision space. All the surrogate models are then coupled with the multi-objective genetic algorithm, and multiple-realisation optimisation is performed incorporating the reliability criterion. This approach results in the robust optimisation of the groundwater management strategies under parameter uncertainty. On validating the derived optimal solutions with the numerical simulation model for different realisations of the uncertain parameters, it was observed that these solutions are robust for the range of values of the uncertain parameters considered. For performance evaluation, the methodology is applied to an existing well field in a realistic coastal aquifer system in the Lower Burdekin in Australia.

Compliance monitoring is an essential component of any groundwater management project. A methodology is developed for the design of compliance monitoring networks in this study. The network is necessary to monitor the compliance of the actual field level implementation with the simulated results. The design is performed subject to the imposed constraint of budgetary limitation, implemented as the maximum permissible number of monitoring wells. Subject to this constraint, the design methodology incorporates two goals within a single objective, viz, to place the monitoring wells where there is maximum uncertainty and to reduce the redundancy in monitored information by minimising the coefficient of correlation between the monitored locations. This objective of monitoring network design is compared against the widely used objective of uncertainty maximisation and the advantages are illustrated. The use of compliance monitoring information to sequentially update the coastal aquifer management strategies is illustrated by simulation experiments. A deviation from the prescribed optimal strategy, at any stage during the field implementation, may result in undesirable effects like increased levels of salinity. Based on the compliance monitoring information, the pumping strategies for the subsequent stages of management are modified to compensate for these ill effects. The results of the simulation experiments conducted for the Lower Burdekin aquifer illustrate that sequential updating of the management strategies based on compliance information helps to better achieve the objectives of management.

A coupled simulation-optimisation framework using trained and tested surrogate models based on genetic programming are shown to be computationally efficient tools for developing optimal extraction strategies for coastal aquifer management. The newly developed ensemble surrogate modelling with multiple realisation optimisation has potential applications in deriving reliable and robust strategies for coastal aquifer management under parameter uncertainty. The developed simulation-optimisation methodology for developing optimal pumping strategies, together with the designed compliance monitoring network and sequential updating of the strategies constitute an integrated approach for the management and monitoring of coastal aquifer systems.

Publications produced during PhD candidature

Peer reviewed:

1) Sreekanth, J & Datta, B 2010, 'Multi-objective management of saltwater intrusion in coastal aquifers using genetic programming and modular neural network based surrogate models', *Journal of Hydrology*, vol. 393 (3–4), pp. 245–256.
2) Sreekanth, J & Datta, B 2011, 'Coupled simulation-optimization model for coastal aquifer management using genetic programming based ensemble surrogate models and multiple realization optimization', *Water Resources Research*, vol. 47, W04516, doi: 10.1029/2010WR009683
3) Sreekanth, J & Datta, B 2011, 'Comparative evaluation of Genetic Programming and Neural Networks as potential surrogate models for coastal aquifer management', *Journal of Water Resources Management*, vol. 25, pp. 3201–3218. (doi: 10.1007/s11269-011-9852-8)
4) Sreekanth, J & Datta, B 2011, 'Artificial intelligence based models for the optimal and sustainable use of groundwater in coastal aquifers', *Journal of Advances in Geosciences*, vol. 23, Hydrological Sciences (HS), pp. 211–222.
5) Sreekanth, J & Datta, B 2011, 'Optimal combined operation of production and barrier wells for the control of saltwater intrusion in coastal groundwater well fields', *Journal of Desalination and Water Treatment*, vol. 32, pp. 1
6) Sreekanth, J & Datta, B 2012, Comment on "Artificial neural network model as a potential alternative for groundwater salinity forecasting" by Pallavi Banarjee et al. [Journal of Hydrology vol. 398(2011), pp. 212–220.], *Journal of Hydrology*, http://dx.doi.org/10.1016/j.jhydrol.2011.12.012
7) Sreekanth, J & Datta, B, 'Stochastic and robust multi-objective optimal management of pumping from coastal aquifers under parameter uncertainty', *Journal of Hydrology* (under review).
8) Sreekanth, J & Datta, B, 'Design of optimal compliance monitoring network and feed-back information for saltwater intrusion management in coastal aquifers', *Journal of Water Resources Planning and Management* (submitted).
9) Sreekanth, J & Datta, B, 'Wavelet and cross-wavelet analysis of groundwater quality signals of saltwater intruded coastal aquifers', *Proceedings of the World Environmental And Water Resources Congress*, Palm Springs, California, May 2011, ASCE, doi:10.1061/41173(414)86.

Conference presentations:

1) Artificial intelligence based surrogate models for the optimal management of seawater intrusion in coastal aquifers, AOGS–2010, Hyderabad, India.
2) Optimal management of saltwater intrusion in coastal aquifers using neural network surrogate models and multi-objective optimization, Challenges in Environmental Science and Engineering–2010, Cairns, Australia.

3) Management models for optimal and sustainable utilization of coastal aquifers using linked simulation-optimization, Groundwater – 2010, The challenge of Sustainable development, Canberra, Australia.
4) Optimal management of coastal aquifers using ensemble based genetic programming surrogate models and multi-objective optimization, IPWE 2011, 4th International Perspective on Water Resources and Environment, Singapore.
5) Sreekanth, J & Datta, B 2011, Optimal pumping strategies to control saltwater intrusion in coastal aquifers, Asia-Pacific Coastal Aquifer Management Meeting, October 18–21, 2011, Jeju Island, Korea.
6) Sreekanth, J & Datta, B 2011, Ensemble surrogate modeling based simulation-optimization, December 5– 9, AGU Fall meeting 2011, San Francisco, USA.
7) Sreekanth, J & Datta, B 2011, Modeling the effects of hydraulic control measures for pumping induced saltwater intrusion for a coastal aquifer in Australia, 50th New Zealand Hydrological Society Conference, December 2011, Te Papa Wellington, New Zealand.
8) Sreekanth, J & Datta, B 2011, Integrating long-term goals with short-term reservoir operation optimization, 50th New Zealand Hydrological Society Conference, December 2011, Te Papa, Wellington, New Zealand.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

1.0 Overview

Land areas adjacent to the world's shorelines support large and ever-increasing concentrations of human population, settlements and socio-economic activities, including many of the world's large cities. It has been reported that, as many as 70% of the world's population dwell in coastal zones (Bear & Cheng 1999). Evidently, conservation and sustainable use of the water resources of these areas assume great importance. Coastal aquifers, i.e. aquifers which are hydraulically connected to the sea, are major sources of freshwater in such areas. Due to the hydraulic continuity with the sea, coastal aquifers are susceptible to salinity intrusion caused by a number of factors, the major ones being the unplanned over-exploitation of the groundwater resource and sea level rise due to climate change. Saltwater intrusion occurs in coastal and deltaic regions all over the world, where population density is high and many human activities take place. Saltwater intrusion in coastal aquifers can prevent the beneficial use of aquifers due to the increased salinity of the groundwater thereby causing great economic and environmental loss. Sustainable use of coastal aquifers requires design and implementation of sustainable management strategies. This thesis is aimed at developing methodologies for optimal management of coastal aquifers ensuring sustainable use. Specifically, this includes computationally feasible simulation-optimisation approaches for developing prescriptive models for management; methodology for improving the reliability of prescriptive models; stochastic and robust management strategies; and monitoring network design and feedback information from compliance monitoring.

Saltwater intrusion in coastal aquifers is a highly non-linear and complex process (Bear et al. 1999). Once salinity intrusion occurs, it requires long-term measures incurring huge costs to remediate contaminated aquifers. Hence, carefully planned strategies of groundwater extraction are required to prevent the eventual contamination of the valuable resource. This work develops a set of methodologies using simulation modelling and optimisation techniques for the efficient and economic management and monitoring of coastal aquifers to control or remediate saltwater intrusion.

Saltwater intrusion is defined as the inflow of saline water into an aquifer system. Salinity intrusion occurs due to the movement of seawater towards the freshwater aquifer thereby creating a brackish environment. Near coastal areas, fresh water and seawater maintains equilibrium, with the heavier seawater underlying the freshwater. A diffuse interface exists between them with the density of the water gradually decreasing from the seawater side to the freshwater side (Figure 1.1). The mixing zone or transition zone has varying thicknesses depending on the coastal aquifer environment. Large-scale saltwater intrusion problems occur when the interface between fresh and saline groundwater moves slowly and smoothly in an upward and/or inland direction. This large-scale displacement can be caused by groundwater abstraction, sea level changes, land reclamation and excavation etc.

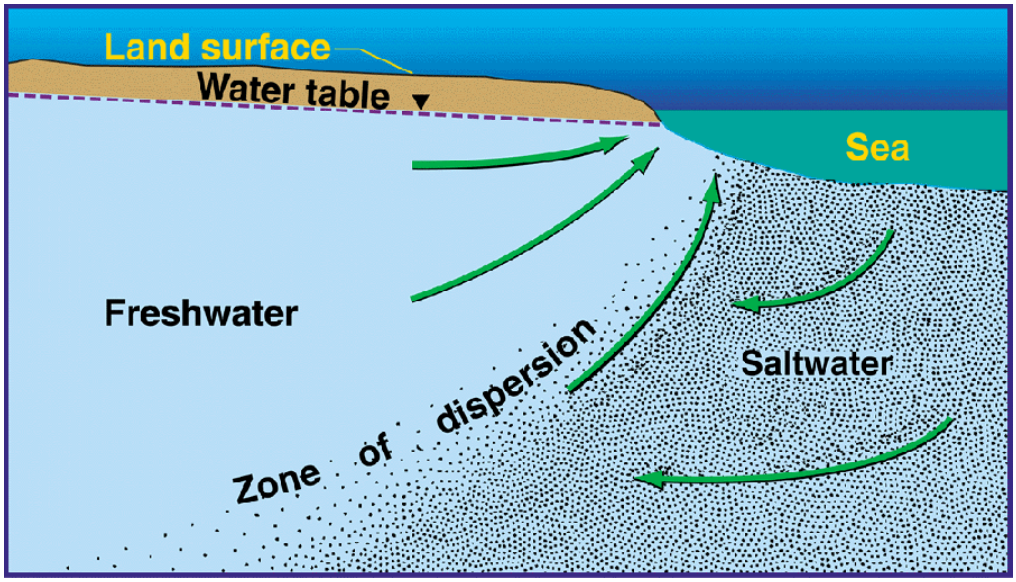

Figure 1.1 Illustration of saltwater intrusion with diffuse interface (USGS, 2010)

A number of management alternatives exist to prevent salinity intrusion. These include demand management, non-potable water reuse, injection barrier, extraction barrier, tapping alternative aquifers, well relocation, plugging abandoned wells, modified pumping rates, pumping caps, physical barriers, scavenger wells, controlled intrusion, conjunctive use, aquifer storage recovery, intrusion with treatment etc. (Maimone 2002).

One of the main reasons behind saltwater intrusion is overexploitation of the coastal aquifers, which disrupts the saltwater-freshwater balance. Hence, the most direct and economical management alternative is to carefully plan and monitor the withdrawal strategies from the coastal aquifer. To study the effects of different withdrawal strategies on the aquifer salinity, the aquifer system needs to be reliably

simulated. Numerical simulation models are used to evaluate the different management alternatives for their effects on the aquifer system. Optimal management strategy cannot be identified by sequentially using the simulation models as there is infinite number of possible alternative withdrawal strategies. Optimisation algorithms are used in a coupled simulation-optimisation framework to identify the optimal withdrawal strategy for coastal aquifers. In coupled simulation-optimisation, each iteration of the optimisation algorithm searches for new and improved withdrawal strategies. In each iteration of the optimisation algorithm, the simulation model is run to quantify the impact of the withdrawal strategies on the saltwater intrusion process. As each simulation model run takes considerable time to execute, the CPU time required for the coupled simulation-optimisation is often very large. In order to reduce this, trained and validated surrogate models like neural networks are used to substitute the numerical simulation model within the optimisation algorithm. This permits the incorporation of any complex simulation model within the optimisation search.

Neural networks have been used in a number of studies to develop surrogate models to approximate density-dependent flow and transport simulation within an optimisation framework (Bhattacharjya & Datta 2005, 2007, 2009). The major drawback in the use of neural network models is the trial and error procedure of determining the neural network architecture and the accompanying large number of input-output connection weights, which adds to the uncertainty in the surrogate prediction model. In this study, an improved surrogate modelling approach using genetic programming is proposed which has specific advantages over neural networks, when used in a coupled simulation-optimisation framework. Genetic programming based surrogate models for approximating the coastal aquifer responses to pumping are developed and coupled with a multi-objective genetic algorithm to derive optimal pumping strategies for coastal aquifer management.

The use of surrogate models to approximate flow and transport processes results in uncertainty in the predictions made, in addition to the uncertainty due to the uncertain parameters of the numerical simulation model. Predictive uncertainty of the surrogate models can adversely affect the optimality of the solutions obtained using coupled simulation-optimisation. A new methodology based on ensemble surrogate models and multiple realisation optimisation is devised to obtain reliable optimal strategies for coastal aquifer management.

In many practical situations, groundwater management strategies need to be developed under poor characterisation of the physical groundwater system. Often, data available for estimation of important parameters like hydraulic conductivity and recharge are sparse and uncertain. In such situations, the stochastic nature of the groundwater parameters needs to be considered and the optimal solutions developed should ideally be robust with regard to the resulting stochastic responses. The methodology based on ensemble surrogate modelling and multiple realisation optimisation is further extended to develop robust and stochastic optimal solutions for coastal aquifer management under parameter uncertainty. The developed methodology is tested for a well field in a realistic coastal aquifer system in the Burdekin region of northern Queensland in Australia.

In spite of developing stochastic and robust optimal strategies for coastal aquifer management, it is still possible that aquifer responses in terms of salinity concentrations may deviate from the predicted values, while the developed optimal strategies are implemented in the field. Evaluating the compliance of the field implementation of any prescribed management strategy is an essential component for any groundwater management project. Optimal design of the compliance monitoring network is required to efficiently and economically collect useful data. A new objective of optimal compliance monitoring network design is therefore introduced. The considered objective function enables locating monitoring wells in regions of maximum uncertainty in terms of the coefficient of variation of concentration and at the same time minimises redundancy in the information collected by avoiding the placement of multiple wells if there is a substantial correlation between the concentrations at the considered monitoring locations.

In practical scenarios, implementation of pumping strategies may deviate from the prescribed optimal strategies due to field level uncertainties. This may lead to diminished benefits from the project and non-compliance of the resulting concentrations with the predicted levels. Ideally, the compliance of the implemented strategies with the prescribed ones needs to be monitored at each stage of implementation of the strategies. When non-compliance is observed at any stage, the pumping strategies for the future stages need to be modified in order to compensate for the observed non-compliance. This is exemplified using numerical experiments in this study. It is observed that sequential updating of the optimal strategies based on

the compliance information results in better compliance levels at the end of the management time horizon.

1.2 Objectives

The main objectives of the thesis are summarised as follows:

1. Development of computationally efficient coupled simulation-optimisation based prescriptive models for the multi-objective optimal and sustainable management of groundwater extraction in coastal aquifers. In achieving this objective, specific advantages of a potential surrogate modelling approach based on genetic programming over popularly used neural network models is illustrated.
2. Improving the reliability of optimal solutions derived using a surrogate-based coupled simulation-optimisation approach. Ensemble surrogate modelling together with multiple-realisation optimisation is proposed to derive reliable optimal strategies for coastal aquifer management.
3. Development of stochastic and robust optimal solutions for the management of groundwater extraction from coastal aquifers considering parameter uncertainty in the modelling of groundwater flow and transport.
4. Design of an optimal compliance monitoring network for monitoring saltwater intrusion in coastal aquifers and sequential modification of the optimal management strategies after each stage of implementation based on the compliance monitoring information.

1.3 Organisation of the thesis

This thesis comprises of eight chapters, including the present chapter which introduces the broad framework of the coastal aquifer management problem and the specific objectives of the thesis.

Chapter 2 deals with the literature and state-of-the-art approaches to saltwater intrusion management and monitoring.

Chapter 3 develops the methodology of the coupled simulation-optimisation approach for developing optimal extraction strategies for coastal aquifers. The specific advantages of genetic programming based surrogate modelling over the commonly used neural networks are also illustrated.

Chapter 4 develops a methodology for improving the reliability of the optimal solutions derived using the surrogate based coupled simulation-optimisation approach.

Chapter 5 deals with the development of a methodology for stochastic and robust optimal management of groundwater extraction from coastal aquifers considering parameter uncertainty and application of the methodology to a realistic coastal aquifer system.

Chapter 6 deals with optimal compliance monitoring network design methodology. It also illustrates the sequential modification of the optimal strategies based on compliance information obtained from the monitoring network.

Chapter 7 discusses the summary, conclusions and recommendations for future work.

2. LITERATURE REVIEW

2.0 Overview

This chapter presents the review of literature specific to saltwater intrusion in coastal aquifers. Saltwater intrusion has been reported from all parts of the world. In this chapter, a few saltwater intrusion problems reported in Australia and across the world are reviewed. In the past few decades, mathematical models have been developed to simulate the saltwater intrusion process. They are descriptive models which can be solved analytically or numerically to simulate the natural aquifer processes or the aquifer response to any induced stress conditions in terms of pumping and recharge. Aquifer management strategies can be developed using prescriptive models. Different methodologies for developing prescriptive models have been developed in the past. A few studies are available on methodologies for monitoring network design and compliance monitoring of saltwater intrusion. Literature on descriptive and prescriptive models for saltwater intrusion management and compliance monitoring are also reviewed in this chapter.

2.1 Saltwater intrusion in Australia and around the world

According to the Australian Bureau of Statistics (2004), more than 85% of Australians live within 50 kilometres of the coastline of Australia. This establishes both the high stress levels associated with and the importance of the coastal groundwater resources of Australia. Due to the increasing population density near the coastal margins which results in over-extraction of groundwater together with the below average rainfall availability, seawater intrusion has resulted in many parts of Australia. Management options including seawater intrusion monitoring networks have been reported as early as the 1970s (Ball et al. 2001). However, Australia-wide assessment of coastal groundwater resources for irrigation in purview of seawater intrusion was first reported by Nation, Werner and Habermehl (2008). Comprehensive investigations on seawater intrusion in coastal aquifers have been carried out for agriculturally important regions of Queensland, Australia like the Pioneer valley, Lower Burdekin, Burnett regions, Bribie Island, Stradbroke Island, Pimpama coastal plain etc. (Werner 2010). Regional-scale saltwater intrusion models have been developed for these regions (Werner & Gallagher 2006; Liu et al. 2006; Narayan,

Schleeberger & Bristow 2007; Werner, 2010). In South Australia, Martin (1997) reported the seawater intrusion in Lefevre Peninsular aquifers from which water is pumped from more than 2500 domestic bores. The groundwater levels in this region are still falling and in places it is as low as 20 metres below the mean sea level (Werner 2010). Blair and Turner (2004) developed three-dimensional seawater intrusion models for a coastal aquifer in Perth, Western Australia. Development of prescriptive models for saltwater intrusion management has not been previously reported in Australia.

Saltwater intrusion has been reported from many other parts of the world as well. For example, Brown (1925) studied and summarised coastal aquifer problems in the United States and Europe as early as the first quarter of the twentieth century. Back and Freeze (1983) noted that 100 years after the first American well had been drilled, there was a widespread increase in saltwater intrusion. Pumping-induced saltwater intrusion has been widely reported all over the world in the past four decades. Specific case studies reported in recent years include studies from Hernando County, Florida (Guvanasen, Wade & Barcelo 2000); Savannah Georgia (Kentel, Gill & Aral 2005); Western Long Island, New York, USA (Misut & Voss 2004, 2007); Llobregat delta, Spain (Abarca et al. 2006); Ravenna, Italy (Giambastiani et al. 2006); coastal karstic aquifer in Crete, Greece (Karterakis et al. 2007); Lower Burdekin, Queensland, Australia (Narayan et al. 2007); Andrapradesh, India (Datta, Vennalakanti & Dhar 2009); coasts of Orissa in India (Rejani, Jha & Panda 2009); Satorini, Greece (Kourakos & Mantoglou 2009); Alabama gulf coast, USA (Lin et al. 2009); Rhodope aquifer system, Northeasterm Greece (Petalas et al. 2009); West Japan (Perera et al. 2010); coastal regions of North America (Barlow & Reichard 2010); Western Morocco (Zouhri et al. 2010); coastland near southern Venice lagoon (Teatini et al. 2010); west coast basin of Los Angeles (Reichard & Johnson 2005); north-eastern coastal area of the UAE (Sherif et al. 2011); Laizhou gulf in China (Qi & Qiu 2011); Miami Beach in north-eastern Spain (Haddad & Marino 2011); Southeastern Italy (Cherubini & Pastore 2011); and the coast of Oman (Schuetze, Grundmann & Schmitz 2011).

2.2 Descriptive models for saltwater intrusion

Different modelling approaches available for the simulation of flow and transport in coastal aquifers are a) sharp interface approach and b) diffuse interface approach. The

sharp interface approach is less accurate where the mixing saltwater and freshwater are assumed to be two immiscible liquids. The diffuse interface approach considers the density dependence of the flow and transport. Due to this, flow and transport equations are required to be simultaneously solved. As a result of this coupling of the equations, the density dependent models are highly non-linear. Historical perspective of saltwater intrusion in coastal aquifers is provided by Reilly and Goodman (1985). Books by Holzbecher (1998), Bear *et al.* (1999), and fundamental papers by others provide insight into the subject.

Sharp-interface models are widely used to solve the saltwater intrusion problem. They are based on the Ghyben-Herzberg approach which assumes a sharp interface between the mixing saltwater and freshwater. This approach works satisfactorily only when the mixing zone is very narrow and it fails when freshwater flows into the sea. The sharp interface approach has been used extensively in numerical as well as analytic models in the past due to its simplicity and lesser computational burden. (Henry 1959; Bear & Dagan 1964; Hantush 1968; Schmorak & Mercado 1969; Sikkema & Van Dam 1982; Dagan & Zeitoun 1998a, 1998b; Naji, Cheng & Ouazar 1999; Park & Aral 2004; Mantoglou, Papantoniou & Giannoulopoulos 2004; Mantoglou & Papantoniou 2008). Strack (1976) developed an analytical solution for the saltwater interface subject to ambient flow and a single pumping well. A solution for a multiple well problem was developed by Cheng *et al.* (2000). Park, Cui and Shi (2009) developed design curves for maximising pumping and minimising injection rates by extending the solution obtained by Cheng *et al.* (2000).

Numerical models available for modelling the density-dependent flow and transport are more accurate in solving the saltwater intrusion problem. Different numerical techniques like Finite Difference, Finite Element, Boundary Element, Finite Volume etc. are available. A number of standard models are available for the density-dependent flow and transport simulations. A review of such codes can be found in Sorek and Pinder (1999). Most commonly used codes are SUTRA (Voss 1984; Voss & Provost 2002), FEMWATER (Lin et al. 1997), SEAWAT (Guo & Langevin 2002), HST3D (Kipp 1986, 1997), FEFLOW (Diersch 2002), MODHMS (HydroGeoLogic Inc 2002) etc.

2.3 Prescriptive models for saltwater intrusion management

Development of a management model involves integrating a simulation model within an optimisation framework. Different approaches used in the past include a) analytical models b) numerical simulation by using the sharp interface approach c) numerical simulation using density-dependent flow and transport c) embedding approach and d) surrogate model based approach.

Management models with analytical solutions for the saltwater intrusion process were developed by Cheng *et al.* (2000), Mantoglou (2003) and Park and Aral (2004). Numerical simulations using the sharp interface approach have been used by Willis and Finney (1985), Finney, Samsuhadi and Willis (1992), Emch and Yeh (1988), Rao et al. (2004a,b), and Mantoglou, Papantoniou and Giannoulopoulos (2004). Park and Aral (2004) used analytical solutions to optimise well locations and pumping rates. Mantoglou and Papantoniou (2008) developed an optimal design for pumping networks in coastal aquifers using sharp interface models. Shi et al. (2011) used a numerical sharp-interface model to determine saltwater and freshwater withdrawal rates at a pumping well.

A number of studies have used density-dependent flow and transport models for developing management models (Das and Datta 1999a,b; Qahman et al. 2005). Das and Datta (1999b) used an embedding approach to develop the management model. Dhar and Datta (2009a,b) used the density-dependent flow and transport simulation model FEMWATER for developing a multi-objective management model. Abd-Elhamid and Javadi (2011) integrated the genetic algorithm optimisation technique with a three-dimensional density-dependent flow and transport simulation model to develop management scenarios to control saltwater intrusion. In the past few decades, simulation-optimisation approaches have been developed as potential tools for solving coastal aquifer management problems.

Initially, classical non-linear optimisation techniques were used to solve most of these optimisation problems. Most of these algorithms are based on random search or gradient-based search techniques. Das and Datta (1999a,b) used the reduced gradient algorithm, MINOS, to solve the saltwater intrusion management problem. In recent years, non-traditional optimisation techniques have gained popularity due to their ability to converge to near global optimal solutions. Evolutionary algorithms and genetic algorithms have been used to solve groundwater management problems. (Cheng et al. 2000; Bhattacharyja and Datta 2005). Genetic algorithm (GA) was used

to solve a multi-objective management problem by Dhar and Datta (2009a). Simulated annealing was used as the optimisation algorithm for coastal aquifer management by Rao et al. (2003, 2004a,b, 2006, 2007a,b). Kourakos and Mantoglou (2009) utilised the Evolutionary Simplex Scheme for pumping optimisation in a coastal aquifer. Sedki and Ouazar (2011) applied a genetic algorithm in conjunction with MODFLOW to explore optimal pumping schemes for a coastal aquifer. Continuous elitist ant colony optimisation was employed by Ataie-Asthiani and Ketabchi (2011) to explore optimal control variable setting for coastal aquifer management. Kourakos and Mantoglou (2011) solved a multi-objective coastal aquifer management problem using the multi-objective genetic algorithm, NSGA-II. Papadopoulou (2011) reports a review of different optimisation approaches applied in developing management methodologies for the control of saltwater intrusion in coastal aquifers. Haddad and Marino (2011) used Honey-bee mating optimisation in combination with sharp interface models to develop optimal management strategies for coastal aquifers.

2.4 Surrogate models in groundwater management

A number of studies have used surrogate models to substitute the numerical simulation model to reduce the computational complexity. The earliest form of surrogate model is the response matrix approach. Response matrices are simplified linear approaches to simulate the aquifer responses. Response matrix approaches have been used by Hallaji and Yazicigil (1996), Zhou, Chen and Liang (2003), Abarca et al. (2006) etc. Complex surrogate models like Artificial Neural Networks have been used more recently due to their capability to model highly non-linear functions.

2.5 Neural networks as surrogate models

Artificial Neural Networks (ANN) have been widely used as surrogates for groundwater models (Ranjithan, Eheart & Garrett 1993; Rogers, Dowla & Johnson 1995; Aly & Peralta 1999). Substantial research work has been done on using Artificial Neural Networks as surrogate models for simulation-optimisation studies. Rao *et al.* (2004a, 2004b, 2006, 2007a, 2007b), Bhattacharya and Datta (2005, 2009), Bhattacharjya, Datta and Satish (2007), Kourakos and Mantoglou (2009) and Dhar and Datta (2009a, 2009b) have used Neural Network surrogate models for developing salinity intrusion management models. Arndt et al. (2005) developed a neural network surrogate model implementing search interval adaptation. The adaptive neural

network model was used as a surrogate for a finite element groundwater model and was used with an optimisation algorithm to solve an optimal design problem. Yan and Minsker (2006) developed an Adaptive Neural Network Genetic Algorithm (ANGA) where the network was trained with search interval adaptation and a genetic algorithm used to solve the optimisation model. Nikolos et al. (2008) developed neural network surrogate models for simulating the groundwater heads within an optimisation framework to determine optimal pumping rates. Behzadian *et al.* (2009) used adaptive neural networks in combination with multi-objective genetic algorithm NSGA-II to locate pressure loggers for a stochastic sampling design.

Kourakos and Mantoglou (2009) developed a Modular Neural Network (MNN) with a number of sub-networks replacing a global ANN. Salinity concentration in each monitoring well was predicted using a modular neural network and the intrusion controlled by relatively few pumping wells falling within a certain control distance from the monitoring wells. The networks were trained adaptively as optimisation progressed. The computational time could be reduced considerably by using the modular neural networks. Papadopoulou, Nikolos and Karatzas (2010) developed a radial basis function artificial neural network based surrogate model in combination with differential evolution for saltwater intrusion management.

One of the main disadvantages in using neural networks in developing the surrogate models is the complexity in functional approximation. With the increasing number of inputs and outputs, the number of connection weights which connect the inputs to the outputs in the neural network architecture increases quadratically. The connection weights are essentially the parameters used in the surrogate models. For any mathematical model, the uncertainty in the predictions made is proportional to the number of parameters used in the model. Hence, as the number of connection weights increases, the predictive uncertainty of the neural network surrogate model also increases. Also, the neural network architecture is most often developed by trial and error in which a number of alternatives are considered and one among them is chosen as the best architecture for functional approximation. Training neural networks for complex functional approximation with a large number of inputs and outputs also incurs a considerable computational burden.

2.6 Genetic programming as a potential surrogate modelling tool

A few studies in the broad area of hydrology and water resources have used GP models (Dorado et al. 2003; Makkeasorn, Chang & Zhou 2008; Parasuraman & Elshorbagy 2008; Wang et al. 2009). GP has been used to develop prediction models for run-off, river stage and real-time wave forecasting. (Babovic & Keijzer 2002; Sheta & Mahmoud 2001; Gaur & Deo 2008). Zechman *et al.* (2005) developed a GP based surrogate model for use in a groundwater pollutant source identification problem. The chemical signals at the observation wells were used to reconstruct the pollution loading scenario. The inverse problem was solved using a simulation-optimisation approach using GA to conduct the search. The numerical model was replaced by a surrogate model developed using genetic programming to reduce the computational burden. From the limited number of studies in the broad area of hydrology and water resources, GP seems to be a simple and efficient tool for functional approximation. Sreekanth and Datta (2010, 2011a, 2011b) developed genetic programming based surrogate models to substitute three-dimensional density-dependent flow and transport simulation models for simulating pumping induced sa1twater intrusion processes within an optimisation framework.

2.7 Surrogate modelling under groundwater parameter uncertainty

Although different surrogate modelling approaches for groundwater management exist, only a few studies have dealt with stochastic surrogate modelling. Aly and Peralta (1999) used neural networks in a stochastic groundwater setting to solve optimal pump-and-treat design problem. In their approach, hydraulic conductivity was considered as uncertain and different realisations of it were generated to train an ANN based surrogate model which predicts a single value of pollutant concentration. The neural network is trained to predict the worst-case scenario of concentration and a constraint is imposed on this value in the optimisation model. Bau and Mayer (2008) developed stochastic management strategies for pump-and-treat-design using surrogate functions. He *et al.* (2010a, 2010b) used a set of proxy simulators in a simulation-optimisation framework to solve a groundwater remediation design problem considering uncertainty in the parameters of the proxy simulators. The residuals in the prediction by the proxy simulators were considered as stochastic and their deterministic equivalent was incorporated into the optimisation model using

chance constraints. Yan and Minsker (2011) developed dynamic surrogate models to replace Monte Carlo simulations within a genetic algorithm to optimise remediation designs.

Surrogate modelling has been widely used to reduce the computational burden on simulation-optimisation approaches for groundwater management. However, there exists little reported research on the use of surrogate modelling based simulation-optimisation in a stochastic groundwater setting.

2.8 Stochastic optimisation in groundwater management

Different stochastic optimisation techniques have been used in the past for optimal decision making under uncertainty (Wagner & Gorelick 1987; Tiedeman & Gorelick 1993; McPhee & Yeh 2006). Chance-constrained programming had been used in groundwater management by Wagner and Gorelick (1987, 1989), Morgan, Eheart and Valocchi (1993), and Datta and Dhiman (1996). Another method for stochastic simulation optimisation is the multiple-realisation approach (Wagner & Gorelick 1989; Morgan, Eheart & Valocchi 1993; Chan 1993; Feyen & Gorelick 2004, 2005; Bayer, Buerger & Finkel 2008). In this method, numerous realisations of uncertain model parameters are considered simultaneously in an optimisation formulation. Recent studies reporting stochastic simulation-optimisation approaches include Bayer, Buerger and Finkel (2008), Singh and Minsker (2008), He *et al.* (2008), Bau and Mayer (2008), Kourakos and Mantoglou (2008), He, Huang and Lu (2009), Ko and Lee (2009), Qin and Huang (2009) and Parker *et al.* (2010). He, Huang and Lu (2010a, 2010b) used a set of proxy simulators, in a coupled simulation-optimisation model for groundwater remediation design under parameter uncertainty of the proxy simulators. The proxy simulators were based on step-wise response surface analysis. The residuals in prediction were treated as stochastic variables and their deterministic equivalent was incorporated into the optimisation model.

2.9 Monitoring network design

Monitoring is important for any groundwater management project. Monitoring network design assumes greater importance due to the cost involved in the water quality monitoring and the inherent uncertainty in the contaminant plume movement. A comprehensive review of saltwater intrusion monitoring in coastal aquifers is given in Melloul and Goldenberg (1997). From their experience in Israel, they recommend a combination of geo-physical methods and direct well observations as an optimal

means of assessing seawater intrusion in coastal aquifers. Monitoring concentrations by direct well observations is expensive as implementation of new wells and periodic sampling and analysis are involved. Optimal monitoring of well networks is required to efficiently and economically monitor saltwater intrusion and the compliance with the optimal strategies of management.

Loaiciga *et al.* (1992) report a comprehensive review of monitoring network design. Zhang, Pinder and Herrera (2005) classified the quantitative approaches for monitoring network design into three broad classes: (1) stochastic simulation approaches (Ahlfeld & Pinder 1992; Massmann & Freeze 1987a, 1987b; Meyer & Brill 1988; Meyer et al. 1988; (2) variance-based approaches (Rouhani 1985; Graham & McLaughlin 1989a, 1989b; Van Geer, Testroet & Zhou 1991; Bierkens, Knotters & Hoogland 2001; Herrera & Pinder 2003) and (3) optimisation-based approaches.

Many optimisation-based approaches have been reported in the literature. Different objectives of optimisation for designing monitoring network include minimisation of variance (McKinney & Loucks 1992; Asefa et al. 2004, 2005; Nunes et al. 2004a, 2004b, 2004c; Heerera & Pinder 2005; Ammar et al. 2008; Chadalavada & Datta 2008; Dokou & Pinder 2009; Ruiz-Cardenas, Ferreira & Schmidt 2009; Chadalavada, Datta & Naidu 2011), contaminant detection (Massmann & Freeze 1987a, 1987b; Meyer & Brill 1988; Hudak & Loaiciga 1992, 1993; Datta & Dhiman 1996; Mahar & Datta 1997; Storck et al. 1997; Montas et al. 2000; Reed, Minsker & Valocchi 2000; Reed & Minsker 2004; Dhar & Datta 2007; Kollat, Reed & Kasprzyk 2008; Bashi-Azghadi & Kerachian 2010), minimisation of monitoring cost (Reed, Minsker & Valocchi 2000; Reed, Minsker & Goldberg 2003; Nunes et al. 2004a; Reed & Minsker 2004; Wu et al. 2006; Kollat & Reed 2007; Kollat, Reed & Kasprzyk 2008; Kollat, Reed & Maxwell 2011), minimisation of mass estimation error (Montas et al. 2000; Reed & Minsker 2004; Wu, Zheng & Chien 2005; Wu et al. 2006; Kollat & Reed 2007) etc. Masoumi and Kerachian (2010) applied an entropy theory for the redesign of optimal groundwater quality monitoring networks.

Dhar and Datta (2009c) developed a monitoring network design for saltwater intrusion monitoring. The objective of the monitoring was minimisation of the sum of the normalised absolute deviation between the estimated concentration based on the designed network and the actual contaminant concentration based on different realisations of the management strategy.

2.10 Research motivation

As evident from the review of existing methodologies for coastal aquifer management, there is scope for the development of computationally feasible and efficient simulation-optimisation approaches for multi-objective optimal management of coastal aquifers. Also, the practical utility of surrogate-based simulation-optimisation approaches is limited most often because the reliability of the surrogate modelling in simulation-optimisation has not been scientifically explored. There is scope for an extension of the surrogate modelling based simulation-optimisation for decision making under groundwater parameter uncertainty. When the data available for modelling is limited, the optimal solutions developed need to be robust to cope with the field uncertainty. When prescriptive strategies for coastal aquifer management are developed under parameter uncertainty, it is important to have methodologies for optimal monitoring of the compliance of the implemented strategies with those prescribed. The prescribed strategies need to be sequentially updated to address any non-compliance issues observed in the field. This study is aimed at developing methodologies which address these issues and contribute to the integrated management of coastal aquifers.

3. SURROGATE-BASED COUPLED SIMULATION-OPTIMISATION FRAMEWORK FOR COASTAL AQUIFER MANAGEMENT

A similar version of this chapter has been published and copyrighted in the *Journal of Hydrology*.

CITATION: Sreekanth, J and Datta, D 2010, 'Multi-objective management of saltwater intrusion in coastal aquifers using genetic programming and modular neural network based surrogate models', *Journal of Hydrology*, vol. 393 (3–4), pp. 245–256.

3.0 Overview

One of the main objectives of this study is to develop computationally feasible simulation-optimisation methodologies for prescribing optimal pumping solutions for coastal aquifer management to control saltwater intrusion. Directly linking flow and transport simulation models to optimisation algorithms has been reported to incur a huge computational burden. Dhar and Datta (2009a) reported a run-time of 30 days for solving an illustrative coastal aquifer management problem based on simulation-optimisation where the numerical simulation model FEMWATER was linked directly to a multi-objective genetic algorithm for optimisation. Previous studies have used neural networks as surrogate models to replace numerical simulation models within optimisation algorithms, to develop computationally feasible coastal aquifer management models. This chapter presents simulation-optimisation approaches based on two surrogate modelling techniques, viz, modular neural networks and genetic programming, as potential surrogate models. These surrogate models are then used for substituting 3D density-dependent coupled flow and transport models within a multi-objective genetic algorithm to develop multi-objective optimal pumping strategies for coastal aquifer management.

Surrogate models based on genetic programming (GP) and a modular neural network (MNN) are developed and linked to a multi-objective genetic algorithm (MOGA) to derive the optimal pumping strategies for coastal aquifer management. Two conflicting objectives of management are considered. Both the surrogate models are trained and tested using input-output patterns of pumping and salinity concentrations generated using the numerical simulation model FEMWATER.

Trained and tested surrogate models are used to predict the salinity concentrations resulting at different locations due to groundwater extraction. A two-stage training strategy is implemented for training the surrogate models. Surrogate models are initially trained with input patterns selected uniformly from the entire search space. Then, optimal management strategies based on the model predictions are derived using the optimisation algorithm. A search space adaptation and model retraining is performed by identifying a modified search space near the initial optimal solution based on the relative importance of the variables in salinity prediction. Retraining of the surrogate models is performed using input-output samples generated in the modified search space. Performance of the methodologies using GP and MNN based surrogate models are compared for an illustrative study area. The capability of GP to identify the impact of input variables and the resulting parsimony of the input variables helps in developing efficient surrogate models. The developed GP models have less uncertainty compared to MNN models as the number of parameters used in GP is much less than that in MNN models. Also a GP based surrogate modelling approach was found to be better suited for an optimisation approach using the search space adaptation.

3.1 Density-dependent flow and transport simulation model

The three-dimensional density-dependent flow and transport simulation model FEMWATER (Lin et al. 1997) was chosen to simulate the coupled flow and transport process in the coastal aquifer system. The relevant equations for the density-dependent flow and transport are as follows (Lin et al. 1997);

3.1.1 Flow equation:

$$\frac{\rho}{\rho_o} F \frac{\partial h}{\partial t} = \nabla \bullet \left[\mathbf{K} \bullet \left(\nabla h + \frac{\rho}{\rho_o} \nabla z \right) \right] + \frac{\rho^*}{\rho_o} q \tag{3.1}$$

$$F = \alpha' \frac{\theta}{n} + \beta' \theta + n \frac{dS}{dh} \tag{3.2}$$

where, F = storage coefficient, h = pressure head, t = time, **K** = hydraulic conductivity tensor, z = potential head, q = source and/or sink, ρ = water density at the chemical concentration C, ρ_o = referenced water density at zero chemical concentration, ρ^* = density of either the injection fluid or the withdrawn water, θ = moisture content, α' = modified compressibility of water, n = porosity of the medium, S = saturation.

The hydraulic conductivity K is given by

$$\mathbf{K} = \frac{\rho g}{\mu}\mathbf{k} = \frac{\left(\rho / \rho_o\right)}{\left(\mu / \mu_o\right)}\frac{\rho_o g}{\mu_o}\mathbf{k}_s k_r = \frac{\rho / \rho_o}{\mu / \mu_o}\mathbf{K}_{so} k_r \tag{3.3}$$

where, μ = dynamic viscosity of water at chemical concentration C, μ_o = referenced dynamic viscosity of water at zero chemical concentration, $\mathbf{k}$ = permeability tensor, $\mathbf{k_s}$ = relative permeability or relative hydraulic conductivity, $\mathbf{K_{so}}$ = referenced saturated hydraulic conductivity tensor.

The density dependence on concentration is given by

$$\frac{\rho}{\rho_o} = a_1 + a_2 C \tag{3.4}$$

where, a_1 and a_2 are the parameters used to define concentration dependence of water density and C is the chemical concentration. The present study considered the viscosity to be independent of chemical concentration.

3.1.2 Flow boundary condition:

A no-flow boundary condition is considered on all boundaries other than a constant head boundary for the sea face given by

$$h = h_d(x_b, y_b, z_b, t) \quad \text{on } B_d \tag{3.5}$$

where, (x_b, y_b, z_b) = spatial coordinate on the boundary, B_d = Dirichlet boundary and h_d = head on the Dirichlet boundary.

3.1.3 Transport equation:

$$\begin{aligned} &\theta\frac{\partial C}{\partial t} + \rho_b\frac{\partial S^a}{\partial t} + \mathbf{V}\bullet\nabla C - \nabla\bullet\left(\theta\mathbf{D}\bullet\nabla C\right) = \\ &-\left(\alpha'\frac{\partial h}{\partial t} + \lambda\right)\left(\theta C + \rho_b S^a\right) - \left(\theta K_w C + \rho_b K_s S^a\right) + \\ &m - \frac{\rho^*}{\rho}qC + \left(F\frac{\partial h}{\partial t} + \frac{\rho_o}{\rho}\mathbf{V}\bullet\nabla\left(\frac{\rho}{\rho_o}\right) - \frac{\partial\theta}{\partial t}\right)C \end{aligned} \tag{3.6}$$

where, ρ_b = bulk density of medium, C = material concentration in aqueous phase, S^a = material concentration in adsorbed phase, t = time, $\mathbf{V}$ = discharge, ∇ = del

operator, $\mathbf{D}$ = dispersion coefficient tensor, α' = compressibility of the medium, h = pressure head, λ = decay constant, $m = qC_{in}$ = artificial mass rate, q = source rate of water, C_{in} = material concentration in the source, K_w = first order biodegradation rate constant through dissolved phase, K_s = first order biodegradation rate through adsorbed phase, F = storage coefficient.

The dispersion coefficient **D** is given by

$$\mathbf{D} = a_T \mid \mathbf{V} \mid \delta + (a_L - a_T)\frac{\mathbf{VV}}{\mid \mathbf{V} \mid} + a_m \theta \tau \boldsymbol{\delta} \tag{3.7}$$

where $\mid \mathbf{V} \mid$ = magnitude of $\mathbf{V}$, $\boldsymbol{\delta}$ = Kronecker delta tensor, a_T = lateral dispersivity, a_L = longitudinal dispersivity, a_m = molecular diffusion coefficient, τ = tortuosity.

3.1.4 Transport boundary condition

A Dirichlet boundary condition of constant concentration, given by the following, was assigned to the seaside boundary;

$$C = C_d(x_b, y_b, z_b, t) \text{ on } B_d \tag{3.8}$$

where, C_d = concentration on the Dirichlet boundary and B_d = Dirichlet boundary

A Dirichlet boundary condition was used for all the boundaries for simulating the transport. The seaside boundary salinity concentration was specified as 35 kg/m^3. This represents the average salinity level of oceans.

In the process of salinity intrusion in coastal aquifers, the concentration at any location at any time is a function of the velocity of flow which is obtained from the solution of the flow equation. The velocity of flow depends on the density of the water, and the density, inturn varies with the salinity concentration in space and time. The density dependence necessitates the coupling of the flow and transport equations. The simultaneous solution of these two coupled equations induces high non-linearity in the simulation of density-dependent flow and transport simulation. FEMWATER (Lin et al. 1997) is one of the few available codes which uses a finite element method to solve density-dependent flow and transport. Imposing the grid over irregular shaped study areas is easier with finite element discretisation. Also, FEMWATER was chosen in this study because of the better understanding of the program resulting from the free availability of the source code.

3.2 Coastal aquifer management model

The coastal aquifer management model developed in the present study essentially has two components. The first component is a surrogate model for predicting the salinity levels in the specified monitoring locations, as a result of the groundwater extraction from the aquifer. The second component is an optimisation algorithm based model to evolve optimal management strategies satisfying the imposed managerial constraints and other system constraints.

3.2.1 Multi-objective pumping optimisation formulation

The management model is developed for a coastal aquifer from which water is extracted using a number of production wells and a number of barrier wells for hydraulic control of gradient adjacent to the sea face. The production wells extract fresh water for beneficial use. The barrier wells extract saltwater from near the coast to hydraulically control the saltwater intrusion process. The developed coastal aquifer management model considers two objectives:

(i) Maximisation of total pumping from production wells in the well field over the management time horizon, and

(ii) Minimisation of the pumping from the barrier wells located close to the sea, which are used to control the hydraulic gradient near the sea to reduce the salinity intrusion.

The physical constraints imposed ensure that the salinity in the aquifer is within prescribed limits. The mathematical formulation is given by

$$\textit{Maximize, } f_1(Q) = \sum_{n=1}^{N}\sum_{t=1}^{T} Q_n^t \tag{3.9}$$

$$\textit{Minimize, } f_2(Q) = \sum_{m=1}^{M}\sum_{t=1}^{T} q_m^t \tag{3.10}$$

$$\text{s.t. } c_i = \xi(Q,q) \tag{3.11}$$

$$c_i \le c_{\max} \forall i,t \tag{3.12}$$

$$Q_{\min} \le Q_n^t \le Q_{\max} \tag{3.13}$$

$$q_{\min} \le q_m^t \le q_{\max} \tag{3.14}$$

where Q_n^t is the pumping from the n^{th} production well during t^{th} time period, q_m^t is the pumping from the m^{th} barrier well during t^{th} time period and c_i is the concentration in the i^{th} monitoring well at the end of the management time horizon.

$\xi(\,)$ represents the density-dependent flow and transport simulation surrogate model and constraint (3.11) represents the coupling of the surrogate model with the optimisation model. M, N and T are respectively the total number of production wells, total number of barrier wells and total number of time steps in the management model. Constraint (3.12) imposes the maximum permissible salt concentration in the monitoring well locations. Constraints (3.13) and (3.14) define lower and upper bounds of the pumping from production wells and barrier wells respectively. The management model is implemented by discretising the study area into discrete cells. Each cell may contain none, one, or more than one production or barrier well, which are aggregated together for each cell.

3.2.2 Coupled simulation-optimisation model

The schematic representation of the salinity intrusion management models developed in this study is shown in figure 3.1. Because the search space for the optimal pumping values is multi-dimensional and continuous, the surrogate models developed in the present work are adaptively trained in two stages.

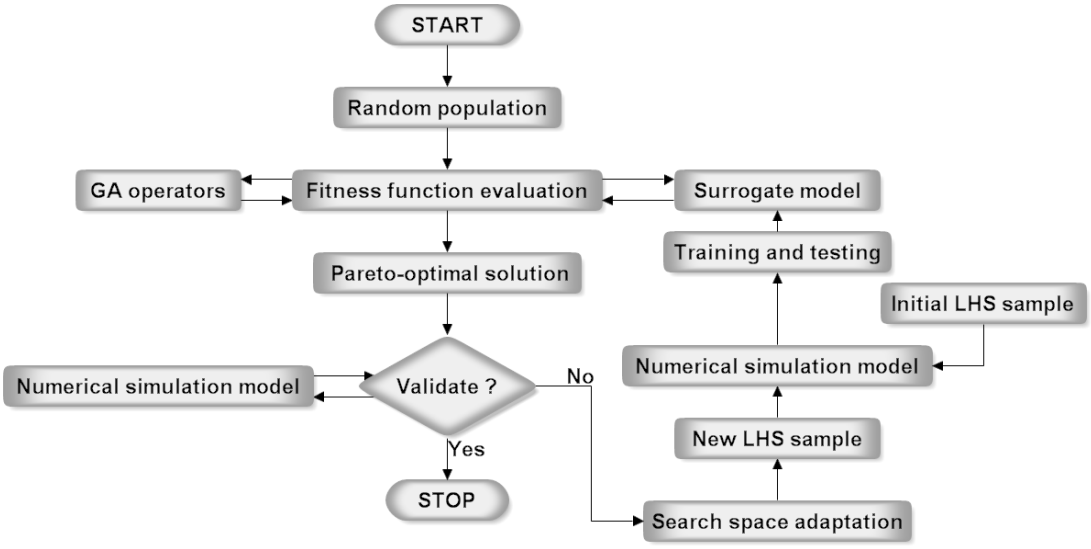

Figure 3.1 Schematic representation of the simulation-optimisation methodology

To train the surrogate models, pumping input values distributed uniformly over the entire search space are generated by Latin Hypercube Sampling (LHS) (Stein 1987). The salinity levels corresponding to each of these patterns are computed using the numerical simulation model, FEMWATER. The surrogate model is trained using input-output patterns of pumping and resulting salinity levels. In the first stage, the surrogate model is trained partially with input-output patterns generated uniformly

over the entire search space. The partially trained surrogate model is linked to the optimisation algorithm to determine the optimal pumping values. The salinity levels resulting due to the optimal pumping are cross-checked using the numerical simulation model. In the second stage, the surrogate model is retrained if the salinity level predictions are not sufficiently accurate. For retraining the surrogate model, training patterns are generated from a reduced search space in the vicinity of the obtained optimal solutions. The search space is selected adaptively based on the spread of the obtained solutions, and the relative importance of the variables in predicting the salinity levels.

3.2.3 GP-MOGA model

Genetic Programming models were developed to predict the salinity concentration at each monitoring well location and were then coupled to the multi-objective genetic algorithm, NSGA II. A brief description of the different components and the implementation GP-MOGA model is presented in this section.

3.2.4 Genetic Programming

Genetic programming (Koza 1994) is used in this study to evolve surrogate models for modelling the salinity intrusion in the coastal aquifers resulting from groundwater abstraction. Genetic programming is an evolutionary algorithm similar to a genetic algorithm in that it uses the concepts of natural selection and genetics in evolutionary computation. For a given model structure and predefined parameter space, a genetic algorithm optimises the parameter values. Genetic programming has an additional degree of freedom which allows an optimum model structure to evolve parallel to optimising the parameter values. Thus, genetic programming identifies the best model structure for simulating the process under consideration while simultaneously estimating the optimal parameter values. Genetic programming learns from examples. The major inputs for the genetic programming model are:

1) Patterns for learning

2) Fitness function (*e.g.* minimising the squared error term)

3) Functional and terminal set

4) Parameters for the genetic operators like the crossover and mutation probabilities.

The functional set consists of the basic mathematical operators and basic functions like addition, subtraction, multiplication, division, trigonometric functions

etc. The choice of the functional set determines the complexity of the model. For example, a functional set with only addition and subtraction results in a linear model structure whereas a functional set which includes trigonometric functions result in highly non-linear model structure. The terminal set consists of constants and variables of the model. The total number of parameters used can be limited to a pre-specified number in order to prevent over-fitting of the model. By using functional and terminal sets, valid syntactically correct programs can be developed. Parse tree notation of two such programs are illustrated in Figure 3.2. Two parent genetic programs are shown in figures 3.2(a) and 3.2(b). The parent programs are crossed over at the dashed sections and the mutation operator changes the value of the constant 2 to 6 to generate two new offspring genetic programs as shown in figures 3.2(c) and 3.2(d).

In the present work, the operators: addition, subtraction, and multiplication are considered in the initial functional set. Later, other functions were added into the functional set one by one in the order of their increasing complexity and non-linearity. For example, an addition or subtraction operation is considered in the functional set before multiplication is considered. However, considering the non-linear nature of the saltwater intrusion process, multiplication and division are considered in the initial functional set. The additional function or operator is accepted upon an improvement in the fitness measure caused by this addition.

GP starts with a set of randomly generated syntactically correct programs. Each program is evaluated by testing the programs in N number of instances, where N is the number of patterns in the training data-set generated by using Latin hypercube sampling and the numerical simulation model. The input-output data set is split into halves. One half is used to train the GP models and the other half is used to test the developed genetic programs. Testing refers to the validation of the model. The testing data set is not used in the fitness function evaluation. Instead, it is used to evaluate how the model performs for a new set of data. Also, the evaluations based on the testing data set are used to pick the best programs from the population.

The fitness value is assigned by comparing the outcome of the program on each of these patterns, with the actual outcome. The fitness function is usually the root mean square error. The programs are ranked based on the fitness value, and new programs are created using the crossover and mutation operators. This process of evolving new programs by means of genetic operators, and subsequent fitness

evaluation is performed for a specified number of generations to obtain the best fit genetic program.

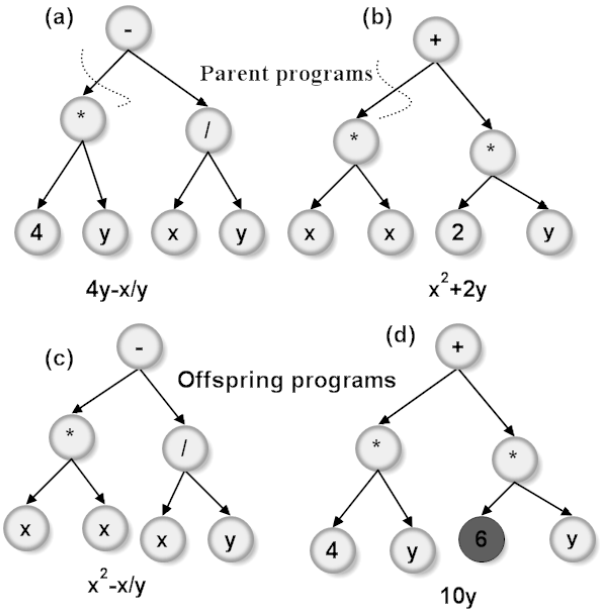

Figure 3.2 Parse-tree representation of parent and offspring genetic programs

The offspring are tested against the fitness function and are retained in the population based on the goodness of fit and are reproduced accordingly. Millions of functions are progressively evolved and tested, mimicking the natural selection process to find the best fit regression models. The present study used Discipulus Genetic Programming software (Francone 1998) for developing the regression models for salinity concentration in monitoring wells. Discipulus uses a Linear Genetic Programming (LGP) algorithm to develop regression models. LGP belongs to the family of GP where multiple calculations are performed on the input variables in a line-by-line fashion. Three quarters of the input-output patterns obtained from the numerical model are used for training and testing the GP model. The rest of the data is used to evaluate the satisfactory application of the model to fresh input values which are not used in the development of the model. The goodness of fit is evaluated using the r-square value.

3.2.5 Multi-objective genetic algorithm NSGA II

A multi-objective genetic algorithm (MOGA) (Deb 2001) is similar to a genetic algorithm (GA) in the initial stages of the search process. It starts the search process with a population of candidate solutions and applies the basic GA operators of crossover and mutation to them to produce the offspring solutions. Selection is

performed on the new set of solutions based on their fitness, to maintain the number of solutions in the population. In addition to these operations, the multi-objective genetic algorithm performs the additional task of organising the members of the population into non-dominated fronts based on the conflicting objectives, and then performs the selection. A solution $x^{(1)}$ is said to dominate the other solution $x^{(2)}$, if both conditions 1 and 2 are true:

1. the solution $x^{(1)}$ is no worse than $x^{(2)}$ in all objectives.
2. the solution $x^{(1)}$ is strictly better than $x^{(2)}$ in at least one objective.

Among a set of solutions *P*, the non-dominated set of solutions P' is that for which any member of P' is not dominated by any member of the set *P*. A non-dominated set of the entire feasible search space S is the globally Pareto-optimal set. Deb (2001) gives a complete description of the multi-objective genetic algorithm. NSGA II was used in this study as it is one of the most widely used multi-objective evolutionary optimisation algorithms.

3.2.6 Linked GP-MOGA model

As in the case of neural network models, Genetic Programming models need to be trained to develop the prediction models. The pumping values for each well for each time period are the inputs to the model. The salinity concentration at each monitoring well location at the end of the management time horizon is the output of each individual GP model. Thus, as many GP models are developed as the number of monitoring wells. Latin Hypercube Sampling (LHS) is used to generate the pumping inputs which are uniformly distributed over the multi-dimensional search space. The set of pumping values along with the resulting salinity concentration at a single monitoring well obtained from the numerical simulation model constitute a single training pattern. The input-output patterns generated are split into three sets: training, validation and applied data sets. The training and validation data set are used for the model development and calibration. The applied data set is a set of independent data which is not used in the model development, but is used to test the applicability of the model.

A random population of candidate programs is initially generated and these programs are operated upon by the operators as mentioned in Section 3.3.1. Addition, subtraction, multiplication, comparison, data transfer and arithmetic operators are used in the programs. The number of parametric constants, which are the parameters

of the GP model, was constrained to be lesser than the number of variables in the case study considered. Sum of the squares of the error was used as the fitness function.

The best programs obtained for the concentration prediction at each of the monitoring wells are validated using the applied data set. This gives the first approximate GP models, based on which the optimisation model is run to obtain the approximate Pareto-optimal solutions. More accurate GP models for the prediction of salinity levels corresponding to the pumping values in the vicinity of these approximate solutions are developed by employing a search space adaptation procedure as described in section 3.4.

3.2.7 MNN-MOGA model

Similar to the GP models, individual modular neural network models were developed for predicting the salinity concentrations at each monitoring well and were linked to the MOGA optimisation model. Each modular neural network predicts the salinity concentration at a single monitoring well. Previous studies have developed neural networks predicting the concentrations at all the monitoring wells using a single global network (Bhattacharya & Datta 2005; Dhar & Datta 2009a). Since the neural networks have a parallel architecture with outputs independent of each other, the global network can be replaced by as many number of sub-networks as the number of outputs of the global network.

Kourakos and Mantoglou (2009) developed a modular neural network (MNN) for salinity prediction with the salinity at a location considered to be influenced by the pumping wells in the proximity of the monitored wells, the proximity being determined by a Hermite interpolation procedure. This methodology permits exclusion of the pumping wells located far away from the monitored location, from being considered as the inputs for the neural network model. The modular neural network model developed in the present study considers all the pumping values to influence the salt concentration at all the locations. This is based on the assumption that at high rates of pumping even the farthest pumping well may influence the salinity at a monitoring location by altering the hydraulic gradient in such a way so as to cause intrusion. However, considering more inputs in the MNN would increase the computational complexity of training the MNN.

MNN models are linked to the multi-objective genetic algorithm NSGA II to derive the non-dominated set of solutions. Based on the approximate network models,

the optimisation algorithm determines the approximate Pareto-optimal front. A search-space adaptation, and model retraining is performed to improve the prediction accuracy of the modular neural network models yielding more precise predictions in the vicinity of the optimal solutions.

3.3 Search space modifications and adaptive training of the surrogate models

The optimisation algorithm searches for optimal solutions in an N-dimensional search space, where N is the number of variables of the problem. Each variable in the optimisation problem adds a dimension to the search space. The search process begins with random evaluations of candidate solutions uniformly distributed over the entire search space. Progressively better solutions are identified and the search process is confined to the space near to the optimal solutions in the variable space. Hence, the surrogate model which predicts the salinity concentrations reasonably well for any combination of inputs in the entire domain is needed in the beginning of the search, and those models which predict the salt concentrations around the near optimal solutions are needed once the approximate solutions are identified. Hence both the GP and MNN models used in this study were trained in two stages. Input patterns uniformly distributed over the entire search space were used for initial training. An expanding set method of training the surrogate models is used, i.e. as the search process progresses more and more training patterns are added to the initial training set based on the direction of search.

Uniformly distributed Latin Hypercube Samples (LHS) of the pumping inputs are generated for the initial training of the surrogate models. The trained and tested surrogate models are used in conjunction with the optimisation algorithm to find the near optimal solutions. Once the near optimal solutions are obtained, a greater of samples are generated in this near optimal space, and the surrogate models are retrained for accurate predictions in the near optimal space.

In the present study, there are multiple optimal solutions (Pareto-optimal front) which are located at distinct locations in the variable space. Hence, the new training set should have statistically generated solutions near all these optimal solutions. The samples for training the new surrogate models are statistically generated from a hypercube around the initial approximate Pareto optimal solutions. The hypercube is generated based on the following two criteria:

1) Length of each side of the hypercube is determined by the spread of the variable that it represents in the Pareto-optimal set.
2) Length of each side of the hypercube is determined by the relative significance of the variable it represents in predicting the salt concentration

Latin Hypercube Samples with uniform distribution are generated from within this hypercube to further train the surrogate models to improve the prediction accuracy near the optimal solutions.

GP has the inherent capability to identify the relative importance of the decision variables and exclude the decision variables with insignificant impact from the model. An impact factor is computed on a 0–1 scale for each of the variables, with a zero impact for the variables which are eliminated from the model. The most important variables have an impact factor of 1. GP evolves thousands of programs to model the input-output relationship. The impact factor of a variable in the GP model refers to the per cent of times that variable has been used in the best 30 models developed by GP. A connection weights method was used to identify the relative importance of variables in the developed modular neural network models. The connection weight method for determining the relative importance of variables is described below.

3.3.1 Connection Weights method – Garson's algorithm

The weights method was developed by Garson (1991) to determine the relative importance of various inputs in a neural network model and had been used in various studies (Goh 1995; Olden & Jackson 2002; Gevrey, Dimopoulos & Lek 2003). The method essentially involves partitioning the hidden-output connection weights of each hidden neuron into components associated with each input neuron. The steps followed are (Gevrey, Dimopoulos & Lek 2003):

1) For each hidden neuron *h*, divide the product of the input-hidden layer weight and hidden-output layer weight by the sum of products of the absolute values of the input-hidden and hidden-output layer weights for all input neurons.

$$P_{i,h} = |W_{i,h}| \times |W_{h,o}| \tag{3.15}$$

$$Q_{ih} = \frac{P_{ih}}{\sum_{i=1}^{ni} P_{ih}} \tag{3.16}$$

2) For each input neuron *i*, divide the sum of the Q_{ih} by the sum for each hidden neuron of the sum for each input neuron of Q_{ih}. The relative importance of all output weights attributable to the given input variable is then obtained. The relative importance is then mapped to a 0–1 scale with the most important variables assuming a value of 1. A RI value of 0 indicates an insignificant variable.

$$RI = \frac{\sum_{h=1}^{nh} Q_{ih}}{\sum_{h=1}^{nh}\sum_{i=1}^{ni} Q_{ih}} \tag{3.17}$$

where, *nh* and *ni* are respectively, the total number of hidden neurons and total number of inputs in the model.

3.4 Evaluation of the developed methodology

The performance of the proposed methodology was evaluated by applying it to an illustrative study area comprising of a portion of a coastal aquifer. Figure 3.3 illustrates the aquifer system with the pumping and barrier wells and the monitoring locations. All the boundaries of the aquifer were taken as no-flow boundaries, except the seaside boundary. The seaside boundary is a constant head and constant concentration boundary with a concentration of 35 kg/m^3. The study area was discretised into triangular finite elements with an average element size of 150 metres. The aquifer was considered to be of a uniform depth of 60 metres.

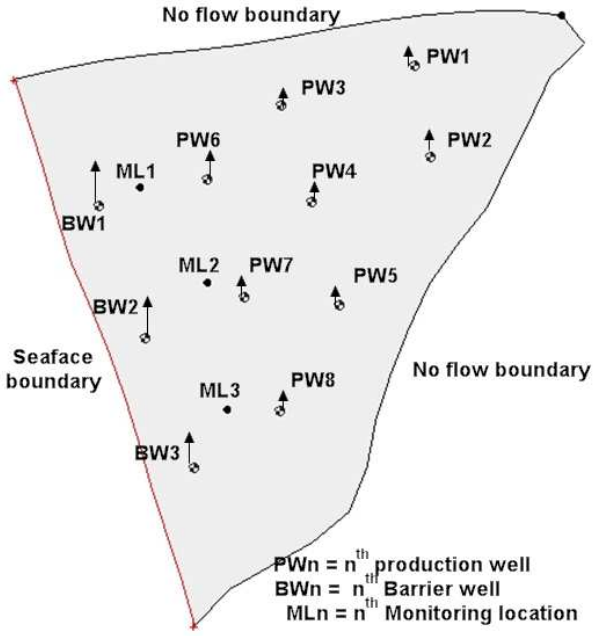

Figure 3.3 Study area with the location of beneficial and barrier wells and monitoring locations

Pumping from 8 potential production well locations and 3 potential barrier well locations were considered. A constant groundwater recharge of 0.00054 m/d distributed over the entire study area was specified. Management time horizon is three years which included three time periods with constant pumping from the wells in each time period. The lower and upper limits of pumping from both pumping and barrier wells were taken as 0 and 1300m^3/d respectively. The number of decision variables is 33 corresponding to pumping from 11 wells over three time periods. The pumping variables are sequentially numbered P1–P33 with P1–P8, P12–P19, and P23–P30 representing the beneficial pumping in the first year, second year, and third year of management respectively. P9–P11, P20–P22 and P30–P33 are the barrier well pumping for the three years of management. The salinity concentrations at the end of the management horizon are represented by C1, C2 and C3. Three monitoring locations were considered, all located in between the barrier and the pumping wells. In the optimisation model, the maximum amount of salinity permissible for C1, C2 and C3 were respectively specified as 0.5kg/m^3, 0.6 kg/m^3 and 0.6 kg/m^3. The U.S. EPA standard for Secondary Maximum Contaminant Level is 0.5 kg/ m^3. Slightly higher values were specified as upper bounds on C2 and C3 to better illustrate the salinity front for the optimal solution.

The management model given in expressions (3.9) to (3.14) has two objectives viz, maximisation of the pumping from the production wells and minimisation of the water extraction from the barrier wells. Barrier wells are installed very close to the sea line. Pumping from the barrier wells induces a steeper hydraulic gradient towards the sea and thus help maintain a flow direction from fresh water to seawater. This helps to prevent the seawater intrusion. These barrier wells act as hydraulic gradient controllers. Thus, increasing the pumping from the production wells requires increasing the pumping from the barrier wells to maintain the salinity levels within the limits. As a result, maximising the pumping from production wells and minimising the pumping from the barrier wells become conflicting objectives.

The equations for the three-dimensional saltwater intrusion process were solved using the numerical code FEMWATER. The concentrations at the monitoring well locations, for all the time periods in the management horizon, were determined by simulating the flow and transport in the aquifer, corresponding to the pumping values. For the initial global training of the surrogate models, 180 patterns of pumping were uniformly generated from the entire search space by using Latin Hypercube Sampling. The salt concentrations at the monitoring locations were obtained from the FEMWATER simulations. Each FEMWATER model takes approximately 10 minutes to run. Approximate models for salinity prediction were developed using both genetic programming and modular neural networks for each monitoring well for each time period. Thus, altogether 18 surrogate models were developed. Of these 9 are GP and the rest are MNN models. However, the models predicting the salinity levels, C1, C2 and C3, at the end of the management time horizon are discussed in the results section.

3.4.1 Model parameters

The parameters used in the numerical simulation model FEMWATER are as follows; Hydraulic conductivity K (isotropic) = 25 m/d, bulk density = 1600 kg/m^3, longitudinal dispersivity = 50 m, lateral dispersivity = 25 m, molecular dispersion coefficient = 0.69 m^2/d, coefficient a_1 = 1 and a_2 = 7.14×10^{-7}.

Data sets with respectively 80, 60 and 40 patterns of pumping inputs and corresponding concentration outputs were used for training, validation and application

of both GP and MNN models. All the genetic programming surrogate models developed used a population size of 500, mutation frequency of 95 and crossover frequency of 50. The operations of addition, arithmetic, multiplication, subtraction, comparison and data transfer were used in the programs to find the regression relation. The maximum number of parametric constants used in the program was limited to 30. Squared deviation from the actual values was used as the fitness function.

The variable space adaptation used for selecting training patterns for the retraining of the surrogate models used the expressions given in (3.18) and (3.19). The domain of the new training patterns and search space was set based on the spread of the solutions in the approximate Pareto-optimal front and the relative importance of each variable. Each Latin Hypercube Sample generated (Stein 1987) to retrain the surrogate models was constrained by the conditions given in (3.18) and (3.19). A Matlab function was used to generate uniformly distributed samples within these bounds.

$$x_i^j \geq Max\left[P_i^{LB}, Min\left((A_i^{opt} - \sigma_i^{opt}), (A_i^{opt} - \sigma_i^{opt}) - I_i \times (P_i^{UB} - P_i^{LB})\right)\right] \qquad \forall i, j$$

(3.18)

$$x_i^j \leq Min\left[P_i^{UB}, Max\left((A_i^{opt} + \sigma_i^{opt}), (A_i^{opt} + \sigma_i^{opt}) + I_i \times (P_i^{UB} - P_i^{LB})\right)\right] \qquad \forall i, j$$

(3.19)

Where A_i^{opt} = the average of the i^{th} variable in the initial Pareto-optimal front, σ_i^{opt} = the standard deviation of the i^{th} variable in the initial Pareto-optimal front, I_i is the maximum relative importance of the i^{th} variable in a 0–1 scale and x_i^j is the j^{th} sample of the i^{th} variable. P_i^{UB} and P_i^{LB} are, respectively, the upper and lower bounds of the i^{th} pumping variable.

All the modular neural network models had one input, hidden, and output layers, each. The number of nodes in the hidden layer was determined by trial and error. A 33-33-1 architecture was found to perform well for the salinity concentrations for each location. Sigmoid activation function was used for all the networks, and training was imparted by a standard back-propagation (BP) algorithm.

The multi-objective genetic algorithm NSGA II used for the pumping optimisation had the following parameters for both GP-MOGA and MNN-MOGA models. The population size and number of generations used were respectively 200 and 1000. The crossover and mutation probabilities were respectively 0.85 and 0.02. These values were fixed after a number of trials using different crossover and mutation probabilities in the prescribed range of values. The pumping optimisation problem had two conflicting objectives with 33 real coded variables and 9 constraints. The distribution indices, i.e. the parameters used to implement simulated binary crossover and mutation in real coded GA (Deb 2001), were respectively 10 and 100.

The developed GP and MNN models were linked to the multi-objective genetic algorithm optimisation model. An optimisation search was performed to obtain the initial approximate Pareto-optimal front. New training patterns were generated near the optimal solutions, as described in section 3.4, to further train the surrogate models. 50 new patterns were generated and the GP and MNN models were retrained with a training set having more density near the optimal solutions. With 180 and 50 training patterns in two stages of surrogate model development a total of 230 patterns were used.

3.4.2 Results and Discussion

GP and MNN models were successfully used to develop efficient surrogates for density-dependent flow and transport simulation models for simulating the salinity intrusion in coastal aquifers. Both these models were particularly advantageous for use in large-scale problems with a large number of pumping and monitoring locations for which training a global neural network surrogate model would be computationally complex. However, in the present work the application of the developed methodology to a small aquifer system to derive multi-objective optimal pumping solutions is illustrated. The use of separate models for predicting the salinity concentrations at each location reduces the model complexity and hence the number of training patterns required to train the models is less as compared to a global surrogate model. Also the use of a modified search space considering the relative importance of the variables further reduces the number of patterns required for the retraining of the models.

The correlation coefficient values (r^2) values for the initial and final GP and MNN models for the salinity predictions at three monitoring locations at the end of the management time horizon are shown in Table 3.1.

Table 3.1 Correlation coefficients for the surrogate models

	Initial stage									Final stage								
	C1			C2			C3			C1			C2			C3		
	Tr	Te	Pr	Tr	Te	Pr	Tr	Te	Pr	Tr	Te	Pr	Tr	Te	Pr	Tr	Te	Pr
GP	0.97	0.94	0.94	0.91	0.83	0.90	0.82	0.92	0.82	0.94	0.97	0.94	0.95	0.94	0.94	0.93	0.91	0.93
MNN	0.98	0.97	0.94	0.93	0.90	0.92	0.90	0.91	0.88	0.93	0.96	0.94	0.94	0.95	0.94	0.91	0.90	0.93

In the initial stage of model development, the MNN model predictions are slightly better than the GP model predictions. The GP models used only 30 model parameters against 1155 weights (parameters of the neural network model) used in the MNN models. However in the final stage, the GP models perform as well as the MNN models, despite the much smaller number of parameters used.

On average, training times for each GP and MNN model were approximately 30 minutes and 36 minutes, respectively. The numerical simulation required approximately 10 minutes for each run. Total time required for the input-output pattern generation using the simulation model FEMWATER was approximately 30 hours. Optimisation in each stage called the surrogate model 200,000 times taking approximately 15 minutes for both GP-MOGA and MNN-MOGA methods to run. Both GP and MNN models are equally capable of generating the output for a given set of inputs within fractions of a second, which substantially reduces the computational burden on the simulation-optimisation model.

An optimisation search was performed using both GP-MOGA and MNN-MOGA methods to find out the approximate Pareto-optimal front of optimal solutions. The Pareto-optimal fronts obtained by the initial GP-MOGA and MNN-MOGA methods are shown in figure 3.4. Each front contains 200 solutions.

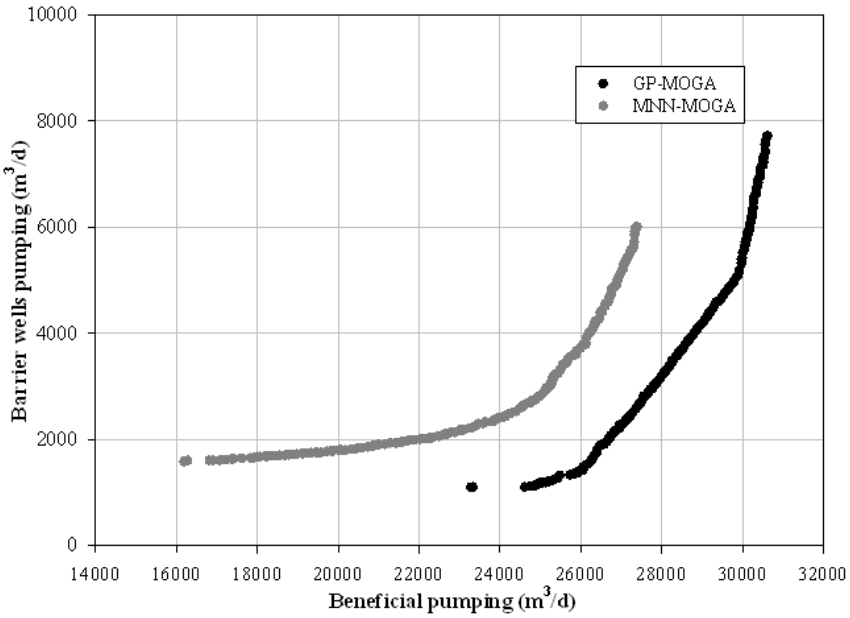

Figure 3.4 Initial Pareto-optimal front

The Pareto optimal fronts obtained by the initial GP-MOGA and MNN-MOGA differ from each other. This is due to the inaccuracy in the prediction of the salinity intrusion by the GP and MNN models near the optimal solutions. Twenty solutions were selected randomly from each of these two Pareto-optimal fronts and were compared with the solutions obtained by using the actual numerical simulation model. Actual numerical simulation of the flow and transport corresponding to the pumping values obtained from these optimal solutions revealed that the GP model predictions of salinity for the optimal solutions did not match the actual values. It was found that GP models consistently under-predicted the salinity values for the optimal solutions obtained after the initial optimisation search. The MNN model predictions had random errors in prediction for each of the three salinity concentrations. This causes the optimal solutions obtained by the GP-MOGA model to actually violate the constraints and move into the infeasible region. The solutions obtained by the MNN-MOGA model were sub-optimal. Thus, the solutions obtained by the initial GP-MOGA and MNN-MOGA models are not truly optimal solutions; instead, they are infeasible or sub-optimal solutions in the vicinity of the actual optimal solutions. These models, therefore, perform the function of identifying a sub-domain of the entire search space where the optimal solutions are located so that the surrogate models can be trained specifically for the input values belonging to this sub-domain.

For the retraining of the surrogate models, the search space adaptation was performed after computing the relative importance of each of the variables. The relative importance of each of the pumping values in predicting the salinity at each monitoring location was computed on a scale of 0 to 1 as mentioned in section 3.4. Relative importance of variables as obtained for the GP and MNN models for salinities C1, 2 and C3 are compared in figures 3.5, 3.6 and 3.7.

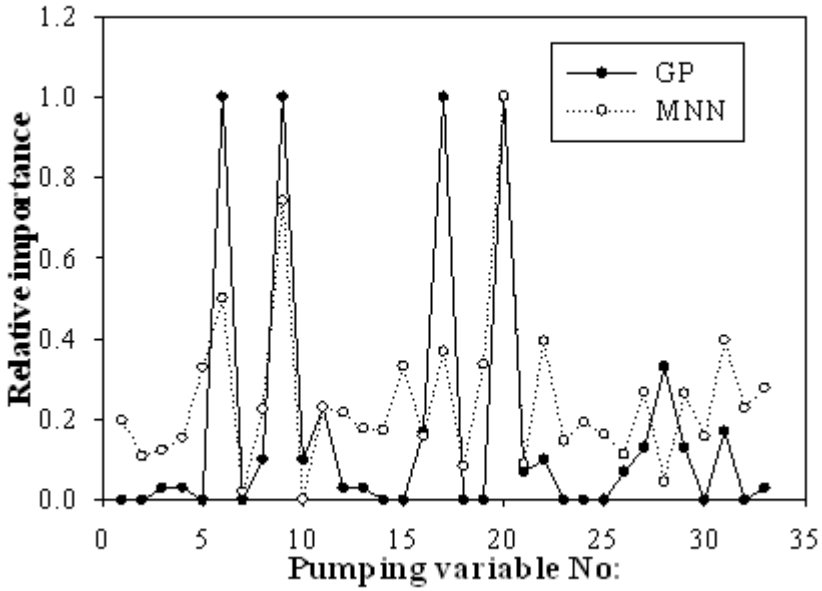

Figure 3.5 Relative importance of pumping variables for predicting C1

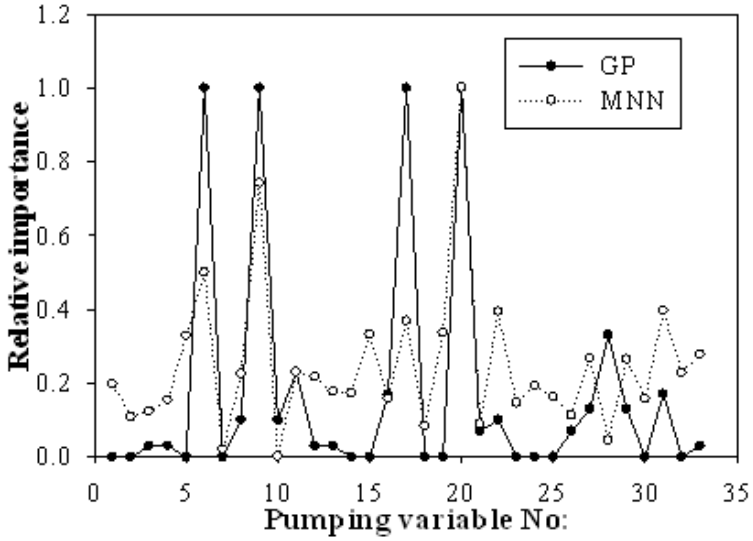

Figure 3.6 Relative importance of pumping variables for predicting C2

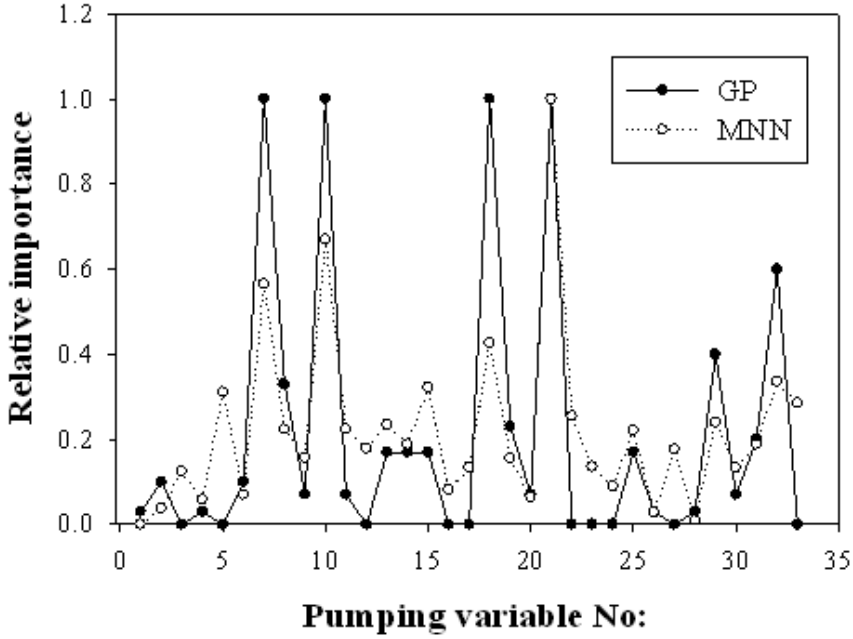

Figure 3.7 Relative importance of pumping variables for predicting C3

Based on the estimated relative importance, all the variables with zero or negligible relative importance are eliminated from the GP models. Thus, in the process of model development, GP selects only those pumping wells which actually contribute to the salinity level in a monitoring location. This elimination of the

insignificant variables helps in faster training, and reduced uncertainty of the models. The parsimonious selection of variables by genetic programming gives it a definite edge over neural networks in developing efficient surrogate models.

Relative importance of the variables was used in the selection of a new training set to retrain the surrogate models. A bigger domain was maintained for the more significant variables. In that way, the lesser significant variables were confined close to the initial optimal values obtained for them, and more significant variables are perturbed about their optimal values obtained from the initial solution. This methodology helps in effectively training the surrogate models by choosing the search-space, and then choosing a training set based on the relative importance of the variables and their spread in the initial Pareto-optimal front.

The variables P6, P9, P17 and P20 are the most significant for the salinity at all the three monitoring locations. It could be observed that genetic programming eliminates the totally insignificant variables from the model. For example, the variable 23 has zero importance for the prediction of concentrations C1, C2 and C3. This implies that the variable 23 (pumping rate from the first well during the third time period) does not contribute to the salinity intrusion when the pumping is within the specified lower and upper bounds. Ideally, the optimal value of this variable is its upper bound which is obtained from the initial management model itself. Therefore, when a refined search space is identified this variable can be confined to the optimal value obtained from the initial solution. The domain for the rest of the variables can be selected proportional to their maximum significance level in the model prediction. This methodology of search space adaptation for retraining of the GP models helps in substantially reducing the number of patterns in the training set generated by using the sampling method.

As an example, a problem with two variables has a two-dimensional variable space as shown in figure 3.8a. Two pumping variables having lower and upper bounds 0 and 1300 respectively are considered. A sampling method should uniformly generate solutions from the 2D space, ideally generating one sample from each dotted square in figure 3.8a, thus requiring 36 samples. However, if the variable Q1 can be identified as insignificant in predicting the salinity levels, then the value of Q1 can be fixed at its upper bound as obtained from the initial optimal solution. Then, only 6

patterns are needed in training the model as seen from figure 3.8b. Similarly, one dimension of the search space can be eliminated for each insignificant variable identified. Therefore, for each significant variable the search space can be modified in proportion to the relative significance of that variable in the model prediction. Also, the search space is modified based on the spread of the solution in the initial Pareto-optimal front as illustrated in figure 3.8c. For example, figure 3.8c illustrates a case where the optimal pumping Q1 is identified in the range 300–600 m^3/d and Q2 in the range 900–1300 m^3/d.

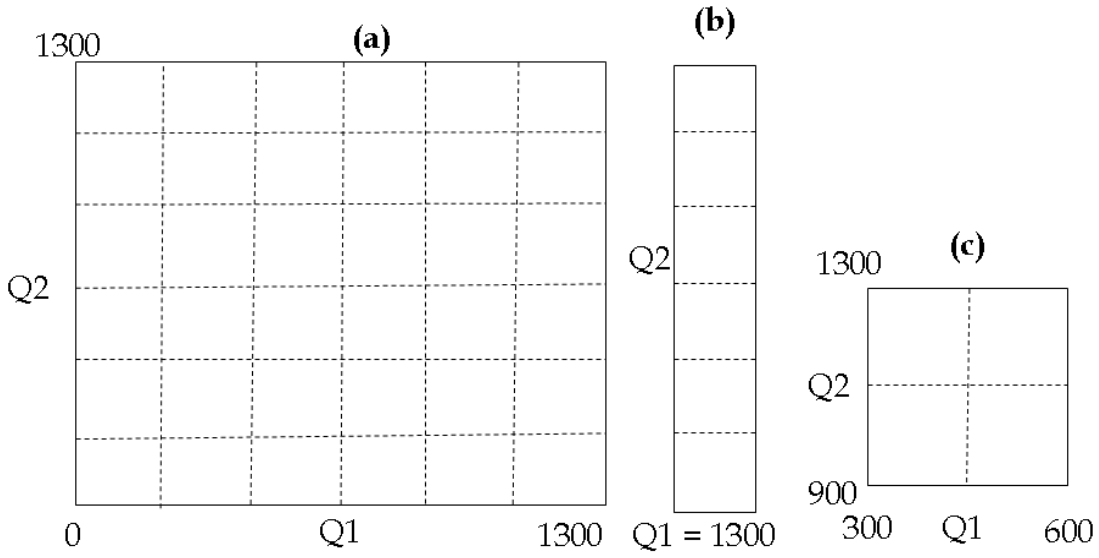

Figure 3.8 2D illustration of search space adaptation

There are 33 variables in the present study with each variable having a domain of 0–1300 m^3/s. This makes the search space a 33-sided hypercube. Thus the number of samples needed to train the surrogate models for a small search space near the initially identified optimal region would be large. The adaptive search space methodology, which modifies the search space with respect to the significance level of the variable, hugely reduces the number of patterns required to train the surrogate model. The GP models identified 13, 2 and 9 variables insignificant in predicting salinities C1, C2, and C3, respectively. This significantly reduces the number of patterns required to train the subsequent GP models. Similarly, a modified search space was identified for the MNN-MOGA model, with each side of the search space proportional to the relative importance (RI) of the variables obtained from the connection weights method.

The final Pareto-optimal solutions provided by both GP-MOGA and MNN-MOGA models are similar. The final Pareto-optimal fronts obtained after retraining the salinity prediction surrogate models with more patterns near the optimal solution are shown in figure 3.9.

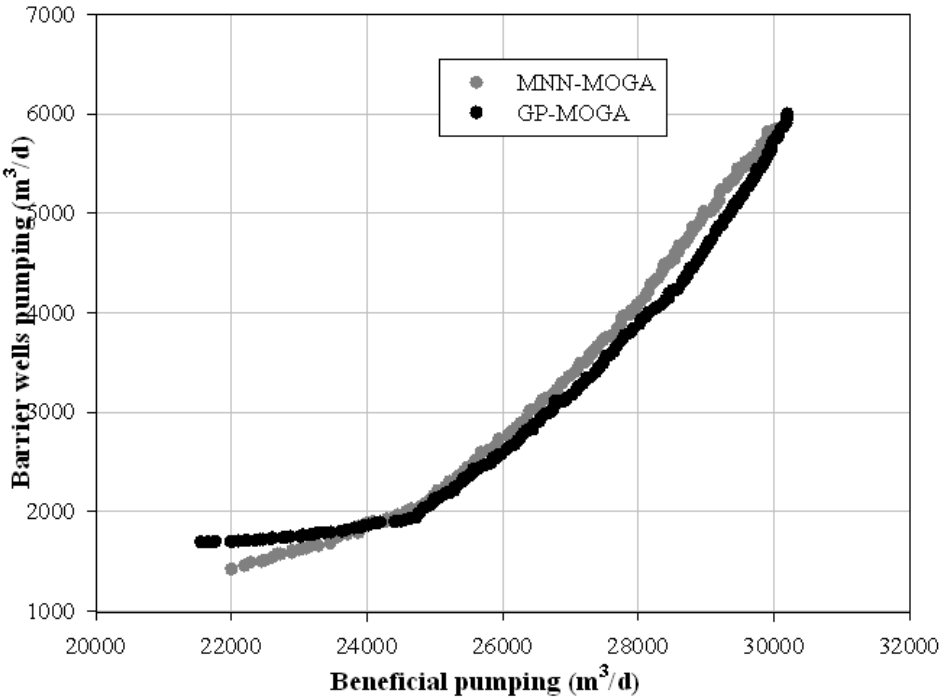

Figure 3.9 Final Pareto-optimal front

The GP and MNN models were retrained with 50 new input-output patterns from the sub-domain where the optimal solutions are located. This increased the prediction accuracy of the models in the vicinity of optimal solutions.

Salinity concentrations at the monitoring locations at the end of the management time horizon predicted by the GP and MNN models for the selected 20 solutions in the front are compared against the corresponding values obtained from the numerical simulation model in tables 3.2 and 3.3. For these selected solutions it was found that the GP predictions were very close to those observed using the FEMWATER model. However, it was found that concentration levels corresponding to some solutions deviated from the surrogate predictions. Hence, it is important that the surrogate-based solutions are validated using the actual numerical simulation model.

Table 3.2 Validation of GP-MOGA model optimal solutions

Solution	C1 ≤0.5 kg/m^3		C2≤ 0.6 kg/m^3		C3≤0.6 kg/m^3	
	GP 10^{-3} kg/m^3	Numerical model 10^{-3} kg/m^3	GP 10^{-3} kg/m^3	Numerical model 10^{-3} kg/m^3	GP 10^{-3} kg/m^3	Numerical model 10^{-3} kg/m^3
1	500.0	500.6	571.8	572.3	599.5	604.2
2	500.0	502.2	573.6	571.2	599.0	595.6
3	500.0	501.1	562.4	563.1	599.7	598.7
4	500.0	497.6	558.1	557.3	599.5	598.5
5	500.0	500.5	558.0	554.8	599.5	598.8
6	500.0	496.5	560.7	562.8	599.7	599.4
7	500.0	499.4	584.9	578.7	599.4	601.7
8	500.0	499.1	575.7	580.3	600.0	600.3
9	500.0	503.5	558.8	565.6	599.7	597.6
10	500.0	504.3	581.1	585.7	600.0	599.8
11	500.0	500.5	559.9	556.7	599.4	599.5
12	500.0	499.8	580.1	576.6	599.6	603.1
13	500.0	502.1	586.5	592.5	599.3	597.4
14	500.0	502.9	561.3	565.9	599.4	599.0
15	500.0	499.6	584.7	587.6	599.6	597.1
16	500.0	495.8	560.8	566.7	599.1	600.9
17	500.0	497.8	560.1	555.3	599.4	597.1
18	500.0	501.7	584.2	578.0	599.9	595.9
19	500.0	504.2	577.2	582.7	600.0	598.8
20	500.0	500.1	560.9	557.1	599.8	600.3

Table 3.3 Validation of MNN-MOGA optimal solutions

Solution	C1 ≤0.5 kg/m^3		C2≤ 0.6 kg/m^3		C3≤ 0.6 kg/m^3	
	MNN 10^{-3} kg/m^3	Numerical model 10^{-3} kg/m^3	MNN 10^{-3} kg/m^3	Numerical model 10^{-3} kg/m^3	MNN 10^{-3} kg/m^3	Numerical model 10^{-3} kg/m^3
1	499.4	500.1	587.9	585.0	599.5	597.3
2	499.9	495.8	587.8	579.8	599.5	601.1
3	499.7	497.1	588.2	588.1	599.9	602.9
4	496.8	499.6	586.5	596.4	599.2	604.1
5	485.5	496.8	559.4	561.3	600.0	599.1
6	499.4	500.2	584.5	583.9	598.1	602.1
7	489.6	502.9	558.5	555.4	599.9	604.0
8	499.0	498.6	595.8	601.7	599.7	602.5
9	499.3	500.0	587.7	593.3	599.0	596.2
10	497.4	502.5	585.2	584.8	596.9	603.2
11	498.4	502.5	598.0	590.5	599.8	603.0
12	496.8	503.5	554.9	546.1	599.5	603.0
13	494.5	496.7	588.8	584.8	598.7	603.4
14	496.5	499.3	595.8	603.9	599.9	599.0
15	496.6	502.6	554.7	558.6	599.6	599.1
16	498.2	498.9	595.0	600.2	599.0	595.6
17	499.6	496.9	595.0	600.5	596.3	604.8
18	499.7	500.6	595.4	596.1	598.5	603.1
19	498.4	499.5	595.7	600.3	599.8	599.7
20	499.0	501.5	551.5	554.6	599.2	602.6

It was found that, for the optimal solutions, salinity concentrations at all monitoring locations were within the specified limits. Also, it was observed that the salinity concentrations C1 and C3 for all the optimal solutions were equal to 0.5 and 0.6 kg/m^3 respectively, i.e. the solutions converge to the upper boundary of the

constraints. GP identifies the relative importance of variables and eliminates the insignificant variables during the model development. Neural network models are incapable of doing this. This study used the connection weights method to identify the relative importance of variables in the MNN model so as to implement the search space adaptation methodology in the MNN-MOGA model. The relative importance of the most significant variables in MNN model as identified by the connection weights method is similar to that in the GP model. However, the flexible model structure of GP enables it to efficiently identify the significant variables, and eliminate the insignificant ones while developing the model.

The neural network models have a pre-set structure having an input layer, hidden layer, and output layer with the number of nodes fixed in each layer. Ideally, an optimised network should give zero connection weights to the neurons connecting outputs to the inputs which are not at all significant in the model. However, this does not occur due to the complex and non-linear nature of the network. Instead, non-zero weights are assigned to all neurons and all the input variables are determinative of the output irrespective of their significance level.

The GP models for salinities C1, C2 and C3 were evolved after evaluating millions of programs, whereas the neural network model optimises the weights of a predefined network structure to obtain best regression fit. Thus GP models are free from the pre-defined model structure. GP uses any combination of the inputs with the permitted operators and model parameters to model the function considered. Thus it has higher degrees of freedom than the neural networks in developing the model. It also does not require the trial and error procedure of determining the optimal model structure. Furthermore, the maximum number of model parameters (constant values used) in all the three GP models were limited to 30, whereas, each of the MNN models has 1155 weights (model parameters) used in predicting the salinity intrusion. The C-code for the GP and MNN models for predicting the salinities at the location C1 is illustrated in appendices A and B respectively. The parameters used in these models can be seen in their respective C-code. Evidently, MNN models have a larger number of parameters and are more complex. As a larger number of parameters contributes to the increased uncertain behaviour of the models, evidently uncertainty levels of GP models are much lower compared those of the neural network models.

GP-MOGA and MNN-MOGA methods applied to the illustrative problem identified identical Pareto-optimal pumping solutions. The surrogate-based salinity intrusion management models are efficient in identifying optimal solutions for the coastal aquifer management problem particularly in reducing the computational burden involved in linked simulation-optimisation methodology. The GP-MOGA method was found to be superior compared to MNN-MOGA in parsimoniously selecting the model variables, thereby increasing the efficiency and decreasing the uncertainty involved in the model. Also, the GP-MOGA method is better suited for the search space adaptation introduced here to train the surrogate models in multiple stages.

3.5 CONCLUSION

Multi-objective pumping optimisation models for coastal aquifers were developed in the simulation-optimisation framework with genetic programming and modular neural networks as surrogate models for simulating the salinity levels. A multi-objective genetic algorithm: NSGA II was used to solve the optimisation problem. Individual surrogate models were developed for predicting the salinity concentrations at each monitoring location, for both GP-MOGA and MNN-MOGA approach. This approach is effective in reducing the computational time and number of training patterns required, compared to a global surrogate model which predicts the salinity levels at all the monitoring locations simultaneously.

The surrogate models were trained in multiple stages with the optimisation search performed after each training stage. Two stages of training and optimisation search were sufficient in the present study to obtain the Pareto-optimal front of solutions. After two stages cross-validation of the surrogate models with the actual numerical simulation models were satisfactory. Also the Pareto-optimal fronts obtained using the two different approaches matched. The methodology can be readily extended to multiple stages of surrogate model training and optimisation for complex problems.

A search space adaptation methodology was implemented to efficiently select training sets for the subsequent stages of model training. The extent of the search space in each dimension was determined by the relative significance of the variable it

represents in the model prediction. Genetic programming is particularly advantageous in implementing this approach as GP can parsimoniously identify the relative impact of the model inputs in the model prediction and select them accordingly.

It is demonstrated that genetic programming based surrogate models can be useful within a simulation-optimisation framework for coastal aquifer management. The characteristic nature of GP in identifying the impact of input variables and the resulting ability of parsimoniously using them in the model development increases the efficiency of the surrogate model. This also reduces the model uncertainty, and is well suited in an optimisation framework. In the illustrative problem, the total number of parameters used in GP was 30 which is much less in number compared to the 1155 parameters used in MNN models. A connection weights method was used to identify the relative importance of variables and implement the same approach in the MNN surrogate model. However, due to the predefined structure of neural network models, insignificant variables could not be eliminated from the prediction model.

To conclude, the contributions described in this chapter can be summarised as:

1) Surrogate model based simulation-optimisation methodologies for the multi-objective management of coastal aquifers were developed. Pareto-optimal solutions were derived using the GP-MOGA and MNN-MOGA management models.

2) A methodology for search space modification and adaptive training of surrogate models was introduced for both GP-MOGA and MNN-MOGA models. Impact of the variables on the model predictions was used to implement the search space modification and adaptive retraining of the models.

3) A genetic programming based approach was found to develop simpler surrogate models which have specific advantages when used in a simulation-optimisation framework. Based on these specific advantages observed for the genetic programming based surrogate modelling approach in simulation-optimisation, the methodologies developed henceforth in this study have utilised GP based surrogate models alone.

4. RELIABILITY MEASURES FOR SURROGATE-BASED SIMULATION-OPTIMISATION APPROACHES FOR COASTAL AQUIFER MANAGEMENT

A similar version of this chapter has been published and copyrighted in the journal: *Water Resources Research.*

CITATION: Sreekanth, J and Datta, B 2011, 'Coupled simulation-optimisation model for coastal aquifer management using genetic programming based ensemble surrogate models and multiple realisation optimisation', *Water Resources Research*, vol. 47, W04516, doi: 10.1029/2010WR009683.

4.0 Overview

As introduced in chapter 3, approximation surrogates can be successfully used to substitute the numerical simulation model within optimisation algorithms to reduce the computational burden on the coupled simulation-optimisation methodology. Surrogate model development comprises of finding out the optimum surrogate model structure and parameter values that relates the inputs to obtain accurate prediction of the outputs. Due to the complex functional relationship between the inputs and output, it is often difficult to find a unique and optimal surrogate model structure and parameter set. The non-uniqueness and uncertainty in the structure and parameters of the surrogate models result in uncertainty in the predictions made by the surrogate model. Thus, while achieving reasonable functional approximations, surrogate modelling results in random errors in the predictions. The reliability of the optimal solutions derived using surrogate-based simulation-optimisation is challenged by the uncertainty in surrogate model simulations. In this chapter, surrogate-based coupled simulation-optimisation methodology is extended and improved for deriving optimal extraction strategies for coastal aquifer management, considering the predictive uncertainty of the surrogate model.

An illustrative coastal aquifer management problem is considered for the new methodology development. Objectives of maximising the pumping from production wells and minimising the barrier well pumping for hydraulic control of saltwater intrusion are considered. Density-dependent flow and transport simulation model

FEMWATER is used to generate input-output patterns of groundwater extraction rates and resulting salinity levels. The non-parametric bootstrap method is used to generate different realisations of this data set. These realisations are used to train different surrogate models using genetic programming for predicting the salinity intrusion in the coastal aquifer. The predictive uncertainty of these surrogate models is quantified. An ensemble of surrogate models is used in a multiple realisation optimisation framework to derive the optimal extraction strategies. The multiple realisations refer to the salinity predictions using different surrogate models in the ensemble. Optimal solutions are obtained for different reliability levels of the surrogate models. The solutions are compared against the solutions obtained using a chance-constrained optimisation formulation and a single surrogate based model. The ensemble-based approach is found to provide reliable solutions for coastal aquifer management while retaining the advantage of surrogate models in reducing computational burden.

4.1 Ensemble of surrogate models

In this chapter the main objective is to develop an ensemble of surrogate models for predicting the saltwater intrusion process in coastal aquifers. Genetic programming is used to develop multiple surrogate models. These models are trained and tested over bootstrap samples of the input-output patterns of pumping rates and salinity levels. The methodology followed in developing the ensemble surrogate models is described below.

4.1.1 Design of experiments

Design of experiments is the first step required for training the GP based surrogate models. Developing a surrogate model based on genetic programming involves learning from input-output patterns. In the case of the coastal aquifer management problem, the inputs are the rates of groundwater abstraction from different potential locations within the aquifer, and outputs are the resulting salinity concentrations. The decision space for the problem under consideration is a multi-dimensional space representing the combinations of groundwater abstraction rates from different locations at various time periods. For the surrogate models to perform satisfactorily, the training patterns should be representative of the entire decision space. Uniformly distributed Latin Hypercube Samples (LHS) of input patterns are generated from the decision space to train the genetic programming based surrogate models.

Latin Hypercube Sampling (LHS), a stratified-random procedure, provides an efficient way of sampling variables from their distributions (Iman & Conover 1982). The LHS involves sampling *ns* values from the prescribed distribution of each of *k* variables X_1, X_2, … X_k. The cumulative distribution for each variable is divided into *N* equi-probable intervals. A value is selected randomly from each interval. The values of *N* obtained for each variable are paired randomly with the other variables.

4.1.2 Non-parametric bootstrap method

A non-parametric bootstrap method is used to generate different realisations of the actual input-output patterns of groundwater abstractions and salinity concentrations. Each realisation of the data set is then used to train a separate surrogate model based on genetic programming. An ensemble of surrogate models for the prediction of salinity levels could be obtained using this procedure. Each surrogate model is distinctly different from the rest in the ensemble, because of the difference in the training data set and the population based optimisation leading to identification of multiple optima by the search algorithm. The distinction in the model structure and parameters amongst the different surrogate models is a manifestation of the uncertainty in the model structure and the parameters themselves.

A methodology used by Parasuraman and Elshorbagy (2008) is followed to accomplish non-parametric bootstrap sampling. The data set obtained using Latin Hypercube Sampling and using the numerical simulation model is assumed to be a representative set of input-output values from the entire population in the decision space. A training data set T of size N is generated using Latin Hypercube Sampling and the numerical simulation model. Different realisations of this data set are obtained using a non-parametric bootstrap method. For this, a bootstrap size of B is chosen. Then B different data sets, each of size N, are obtained by repeated random sampling with replacement from the set T. Thus, each bootstrap sample-set T_B has different input-output patterns from the training data set T repeated many times. The bootstrap sample sets T_B differ from each other only in terms of the repetition of some patterns and the elimination of some patterns from the original data set. The repetition of patterns in the bootstrap causes differential weighting of these patterns. This results in development of the models which are different in their predictive capability in different regions of the decision space of the prediction model. This also triggers the convergence to multiple optimal solutions while training the prediction model. Thus

each surrogate model is an optimal model for the prediction. However, each is different in its predictive capability in different regions of the decision space, depending on the weights assigned to patterns from each region.

The performance of each of the surrogate models is determined by evaluating the root mean square error based on the testing data set. After computing the root mean squared errors for each surrogate model in the ensemble, the standard deviation and coefficient of variation of these errors are computed. The coefficient of variation of these errors is a measure of the predictive uncertainty of the models. The number of surrogate models in the ensemble is determined by performing an incremental statistical analysis on the ensemble performance, i.e. surrogate models are sequentially added in to the ensemble and the resulting uncertainty is evaluated. Also, the RMSE of the resulting ensemble is computed after the addition of each surrogate model. RMSE is computed based on the testing data, considering the testing data sets of all the surrogates in the ensemble taken together at each stage of addition. The optimum number of surrogate models in the ensemble is determined as follows. An ensemble with 10 surrogate models is considered initially. The root mean square error of the salinity concentration predictions by each surrogate model is computed. The coefficient of variation of these root mean square errors is computed and is considered as the measure of uncertainty in the ensemble of models. Then, new surrogate models are added into the ensemble one at a time and the resulting RMSE and uncertainty are computed. This procedure is repeated until there is no significant change in the uncertainty of the ensemble with further addition of surrogate models. The number of surrogate models in the ensemble at this stage is the ensemble size. The number of models in the ensemble at which further addition of models into the ensemble do not produce significant change in the uncertainty is considered as the optimum number of surrogate models in the ensemble.

4.2 Optimisation approaches

The main objective of this study is to develop a coastal aquifer management model which uses an ensemble of surrogate models to simulate the saltwater intrusion process. Two approaches of optimisation addressing the uncertainty in surrogate model predictions are used in this study. The first one is based on a stochastic simulation-optimisation method called a multiple-realisation approach (Gorelick 1987; Wagner & Gorelick 1989; Morgan et al. 1993; Chan 1993; Feyen & Gorelick

2004). The second approach uses a chance-constrained optimisation model (Gorelick 1987,1989; Morgan et al. 1994; Datta & Dhiman 1996).

The stochastic optimisation accounts for the uncertainty in the surrogate model structures and parameters. In the multiple realisation approach all the surrogate models in the ensemble are independently linked to the optimisation model, i.e. if the ensemble consists of 10 different surrogate models then the optimisation formulation has a stack of 10 constraints representing the surrogate models. Thus the optimal solution will be subject to satisfying each of these constraints representing the different surrogate models, which differ from each other due to the model structure and parameter uncertainty.

4.2.1 Multi-objective optimisation using the multiple-realisation approach

A multi-objective optimisation formulation similar to that described in equations 3.9 to 3.14 in chapter 3 is adopted. However, equation 3.11 and 3.12 are modified as follows;

$$\text{s.t. } c_i^r = \xi_i^r(Q,q) \quad \forall i,r \tag{4.1}$$

$$c_i^r \le c_{\max} \forall i,r \tag{4.2}$$

where c_i^r is the r^{th} realisation of concentration in the i^{th} location at the end of the management time horizon. This is obtained from the r^{th} surrogate model for the salinity at i^{th} location using the surrogate model given by $\xi_i^r(\,)$.

With the multiple realisation approach, optimal solutions with different reliability values can be obtained. The reliability value is the fraction of surrogate models in the entire ensemble with salinity predictions that satisfy the imposed constraints of maximum salinity levels in the optimisation model. For example, if there are N different surrogate models in the ensemble, it is possible to obtain an optimal solution with a reliability of n/N by constraining the optimisation model to satisfy constraints imposed by at least n of the total N surrogate models. Reliability of the optimal solution is close to 1 when the constraints imposed by all N surrogate models are satisfied. However, this reliability pertains to the uncertainty in the ensemble of surrogate models only.

4.2.2 Chance-constrained approach

The optimal solutions obtained by the multiple-realisation approach for different reliabilities are compared to the solutions obtained using a chance-constrained optimisation formulation. The chance-constrained formulation uses the same objective functions and constraints as in (3.9) – (3.10) and (3.13) – (3.14). The constraints given by (4.1) and (4.2) are replaced as follows:

$$c_i = \xi_i^{\mu}(Q,q) + \varepsilon_i \tag{4.3}$$

$$Rel\,[c_i(Q,q,\varepsilon_i) \le c_{\max}] \ge \beta \tag{4.4}$$

where, c_i is the salinity concentration at the i^{th} location at the end of the management time-horizon, ε_i is the error in the salinity concentration prediction for the i^{th} location and $\xi_i^{\mu}(Q,q)$ is the average of the salinities at the i^{th} location predicted by the ensemble of surrogate models. *Rel* is the reliability level of the ensemble prediction that the predicted concentration is less than $c_{\max}$. This reliability is based on the cumulative distribution function of the error residuals in the salinity level prediction by the surrogate models. The reliability is constrained to be greater than or equal to β. The probabilistic constraint in (4.4) is converted into its deterministic equivalent as follows:

$$\xi_i^{\mu}(Q,q,\varepsilon_i) + \phi_i^{-1}(\beta) \le c_{\max} \tag{4.5}$$

where ϕ_i^{-1} is the inverse cumulative distribution function for the residuals in salinity prediction at the i^{th} location and $\phi_i^{-1}(\beta)$ gives the prediction error corresponding to a reliability β.

A coupled simulation optimisation model with a single surrogate model predicting the salinity levels at each monitoring locations is also developed for comparative evaluation. The same optimisation formulation as in (3.9) – (3.14) is used for this purpose except that salinity prediction by the ensemble represented by (3.11) is replaced as follows:

$$c_i = \xi_i^{b}(Q,q) \tag{4.6}$$

where ξ_i^{b} represents the best surrogate model, in terms of the least value of the objective function obtained in the GP model, for predicting the salinity at the i^{th} location. The original data set is used to develop this surrogate model instead of the bootstrap sample.

4.2.3 Multi-Objective Genetic Algorithm

The same multi-objective genetic algorithm NSGA-II (Deb 2001), as mentioned in chapter 3, is used to solve the multi-objective coastal aquifer management problem. NSGA-II uses a population of candidate solutions to the optimisation problem with the GA operators crossover, mutation and selection to evolve improved solutions over a number of generations. In addition to this, NSGA-II organises the members of the population into non-dominated fronts after each generation, based on the conflicting objectives of optimisation. Thus, in a single run, NSGA-II is able to generate the entire Pareto-optimal set of solutions at the end of the specified number of generations.

4.3 Ensemble-based coupled simulation optimisation model

The coastal aquifer management model makes use of a coupled simulation optimisation framework to derive the optimal groundwater extraction strategies for coastal aquifers. The ensembles of surrogate models for simulating the aquifer responses in terms of salinity concentrations are coupled with the optimisation model by linking each surrogate model separately with the optimisation algorithm. The multi-objective genetic algorithm randomly generates candidate solutions which are the groundwater extraction rates for the different time periods within the management horizon. The aquifer responses corresponding to each of these patterns of extraction are obtained from the ensemble of surrogate models. All generated candidate solutions are evaluated for feasibility and fitness. New candidate solutions are generated using the genetic algorithm operators. The procedure is repeated for a number of generations, until the termination criteria are satisfied. The solutions are progressively improved to converge to the final Pareto-optimal front. A schematic representation of the ensemble-based simulation-optimisation model is shown in figure 4.1.

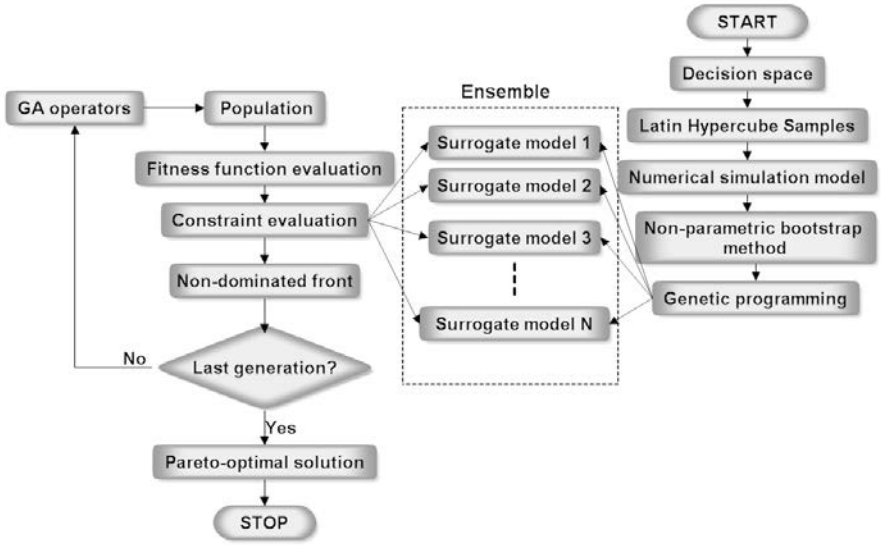

Figure 4.1 Schematic representation of the ensemble-based coupled simulation-optimisation method

4.4 Validation

Once the optimal solution is obtained, its validity is checked by simulating the aquifer processes by using the optimal pumping values in the actual numerical simulation model FEMWATER. The residual in the salinity prediction, i.e. the difference between the surrogate-predicted value and the numerically simulated value is evaluated for five optimal solutions in different regions of the Pareto-optimal front. This is performed for the optimal solutions obtained using the three optimisation models, viz, single surrogate model, ensemble-based model and the chance-constrained model.

4.5 Case study

In order to illustrate the application of the proposed methodology, it is applied to derive optimal extraction strategies for an illustrative coastal aquifer system similar to that described in chapter 3. A three-dimensional view of the aquifer system is illustrated in figure 4.2. The 8 potential locations for beneficial groundwater extraction are shown as PW1–PW8. The barrier well locations for hydraulic control of saltwater intrusion are shown as BW1–BW3. The salinity concentrations were monitored at three locations, C1, C2 and C3, at the end of the management time horizon.

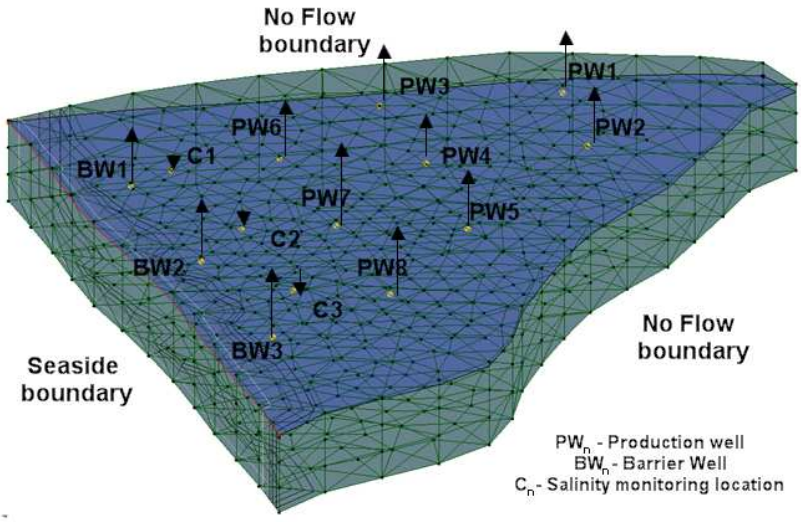

Figure 4.2 Three-dimensional aquifer system illustrating the well and monitoring locations

The time horizon for the management model was fixed as three years with the extraction rates in each management period of one year considered as uniform. The groundwater recharge is specified as a constant rate of 0.00054 m/d, respectively. The lower and upper limits on groundwater abstractions for both beneficial and barrier wells are 0 and 1300 m^3/d. The total number of decision variables in the optimisation model is 33, corresponding to pumping from 11 wells for 3 time periods. The management model specifies a maximum permissible salt concentration limit of 0.5 kg/m^3, 0.6 kg/m^3 and 0.6 kg/m^3 at these locations respectively. The parameters used for the FEMWATER model are the same as those used in chapter 3 and are given in table 4.1.

Table 4.1 Parameters for simulating the aquifer processes

Parameter	Value
Hydraulic conductivity in x-direction	25 m/d
Hydraulic conductivity in y-direction	25 m/d
Hydraulic conductivity in z-direction	0.25 m/d
Longitudinal dispersivity	80 m/d
Lateral dispersivity	35 m/d
Molecular diffusion coefficient	0.69 m^2/d
Soil porosity	0.2
Density reference ratio	7.14x10-7

A three-dimensional coupled flow and transport simulation model was used to simulate the aquifer processes resulting in salinity intrusion due to groundwater abstraction in this study area. Different groundwater extraction scenarios were generated using Latin Hypercube Sampling. The salinity concentrations resulting due to each of these pumping patterns are simulated using FEMWATER. The simulated salinity level and the corresponding pumping rates form the input-output pattern. Altogether, 230 extraction patterns are used in this study. Different realisations of this input-output data set were generated using a non-parametric bootstrap method. Each of these data sets was used to build surrogate models to create the ensemble of surrogate models. Each data set was split in half for training and testing the GP models. The input-output patterns were then used to train the genetic programming based surrogate models.

Surrogates were developed for predicting salinity at three different locations. For each location 30 models in the ensemble were found to be sufficient to characterise the uncertainty. All the genetic programming surrogate models used a population size of 500, mutation frequency of 95 and crossover frequency of 50. Genetic programming software, Discipulus, was used to develop the surrogate models. The parameter values selected as per the guidelines after performing a sensitivity analysis were used in the development of the model. The functional set in the developed GP models contained the following operations – addition, subtraction, multiplication, division, comparison and data transfer. The maximum number of surrogate model parameters used was limited to 30 to prevent over fitting of the model. Squared deviation from actual value was used as the fitness function. At the end of the model training and testing, the source code of the GP model in the C programming language was generated using the interactive evaluator of the software. This C code was then coupled with the multi-objective optimisation algorithm NSGA-II.

4.6 Results and Discussion

4.6.1 Uncertainty in the surrogate models

The uncertainty in the surrogate models was quantified using the coefficient of variation of the root mean square errors (RMSEs) of the individual surrogate models. The root mean square errors of individual surrogate model salinity predictions C1, C2 and C3 are shown in figures 4.3, 4.4 and 4.5. The RMSEs are computed over the

testing data set used for evaluating the genetic programming based surrogate models. It could be observed that for different realisations of the same data set, the root mean square errors are different for different surrogate models. This is due to the predictive uncertainty of the surrogate models. The root mean square errors for the ensemble of models predicting salinity C1 are plotted against the number of surrogate models in the ensemble starting from an initial ensemble size of 10 in figure 4.6**.** In this example, as the number of models in the ensemble increases, the RMSE of the ensemble prediction decreases.

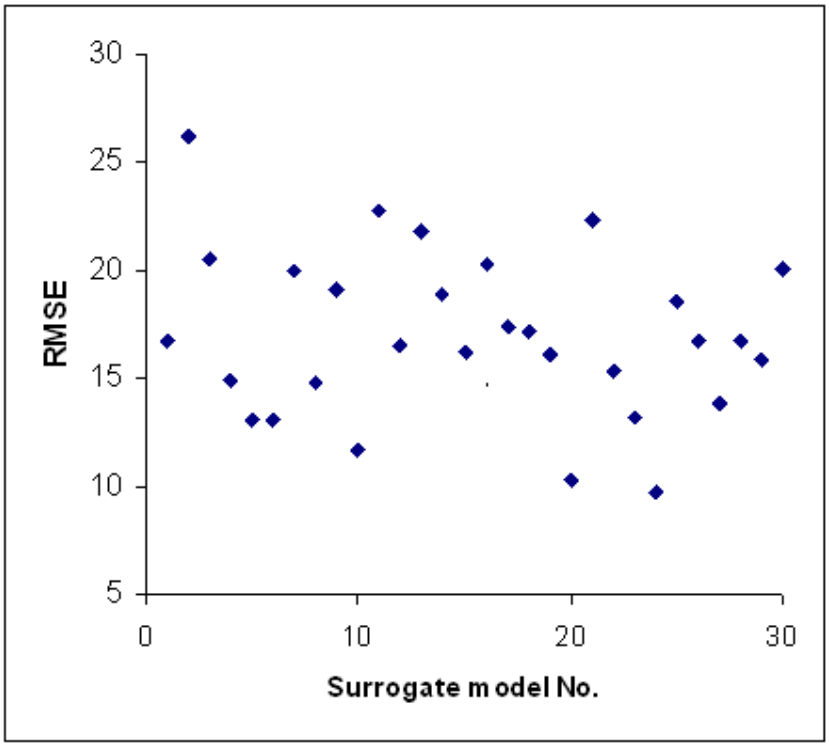

Figure 4.3 RMSE for individual surrogate models simulating salinity C1

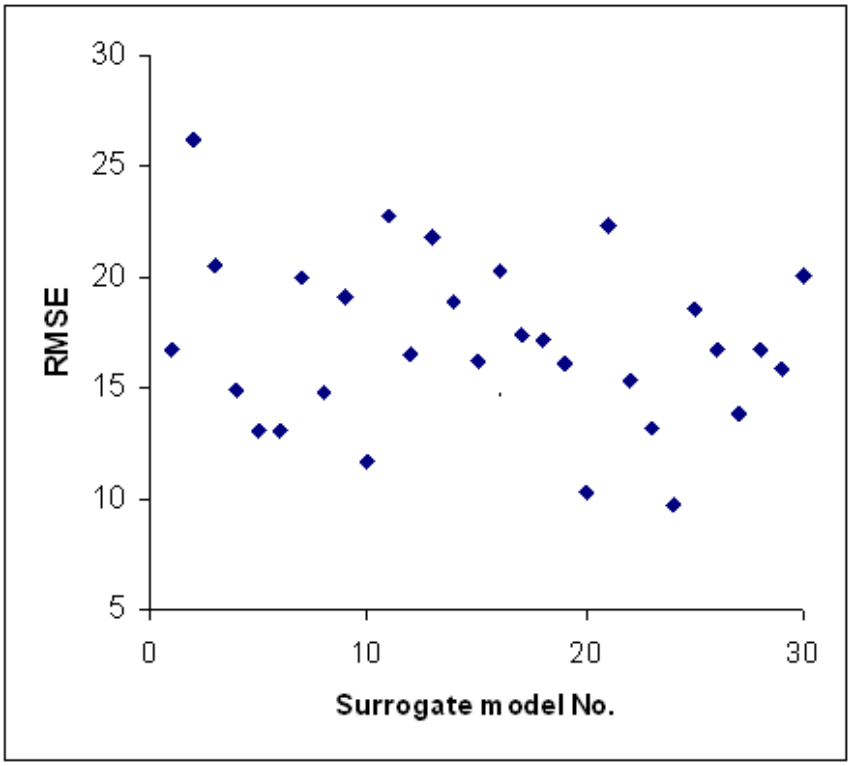

Figure 4.4 RMSE for individual surrogate models simulating salinity C2

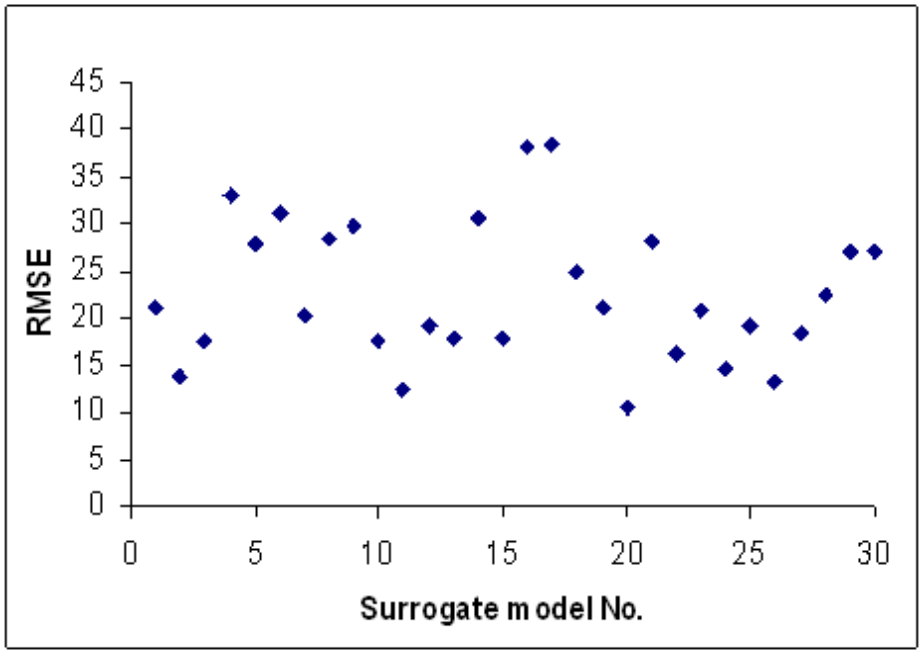

Figure 4.5 RMSE for individual surrogate models simulating salinity C3

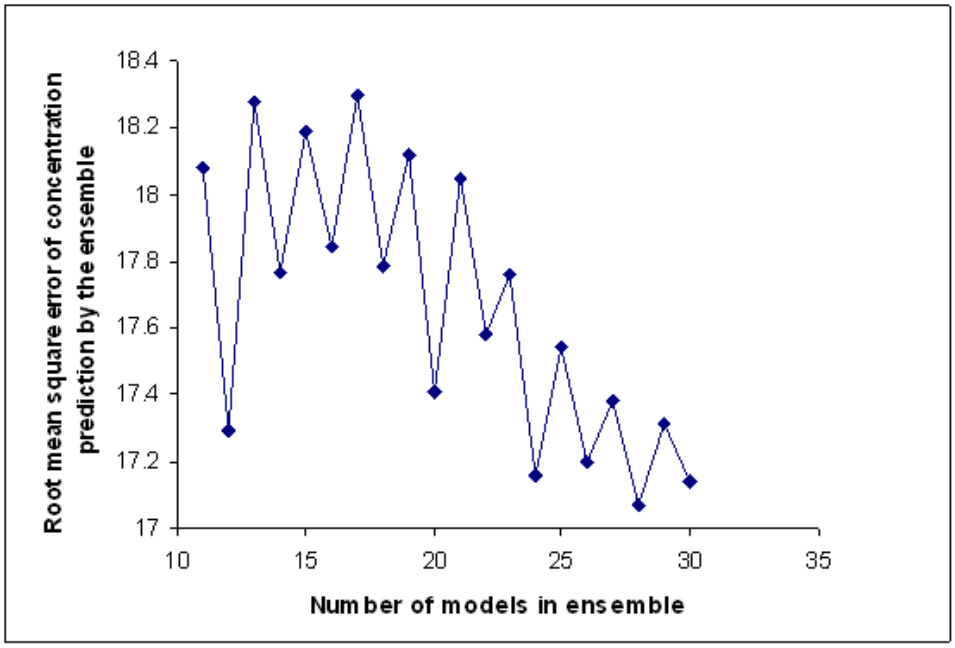

Figure 4.6 RMSE of the ensemble simulating salinity C1

The coefficient of variation of the RMSEs, as a measure of uncertainty in prediction of salinity, is plotted against the number of surrogate models in the ensemble for each ensemble predicting C1, C2 and C3. The plots are shown in figures 4.7, 4.8 and 4.9. Uncertainty of the ensemble model has a definite decreasing trend with the increasing number of models in the ensemble. For each of the salinity concentrations C1, C2 and C3 the uncertainty in the ensemble of the surrogate model decreases with the number of models in the ensemble and reaches a constant value when the number of models in the ensemble is around 30. Hence, the optimum number of models in the ensemble for coupled simulation optimisation is chosen as 30. The optimum number of surrogate models depends on the uncertainty level in the model structure and parameters. For more complex systems, the uncertainty in the model structure and parameters of surrogate models will be larger, and hence larger number of surrogate models will be required in the ensemble. The sensitivity of the derived Pareto-optimal solutions to the number of surrogate models in the ensemble is analysed in section 4.6.4.

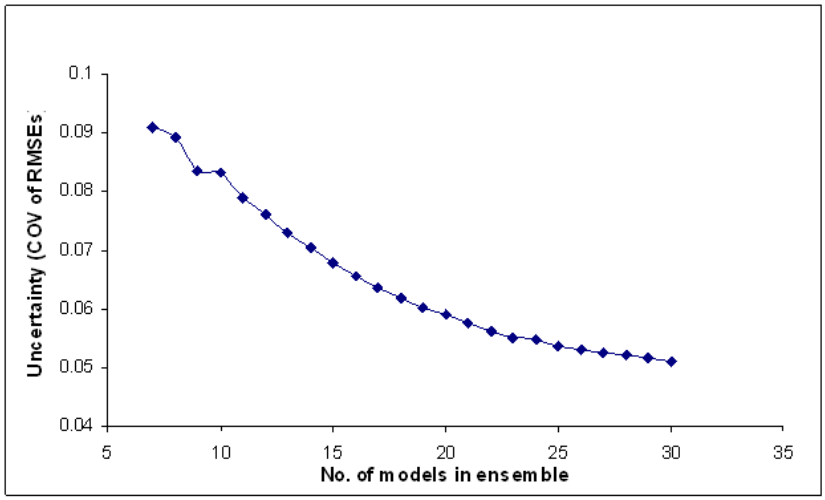

Figure 4.7 Uncertainty levels for increasing ensemble size for salinity C1

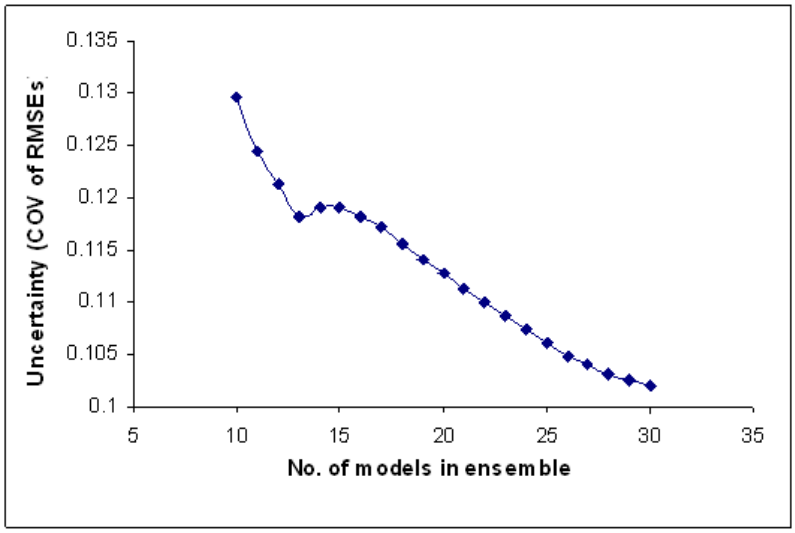

Figure 4.8 Uncertainty levels for increasing ensemble size for salinity C2

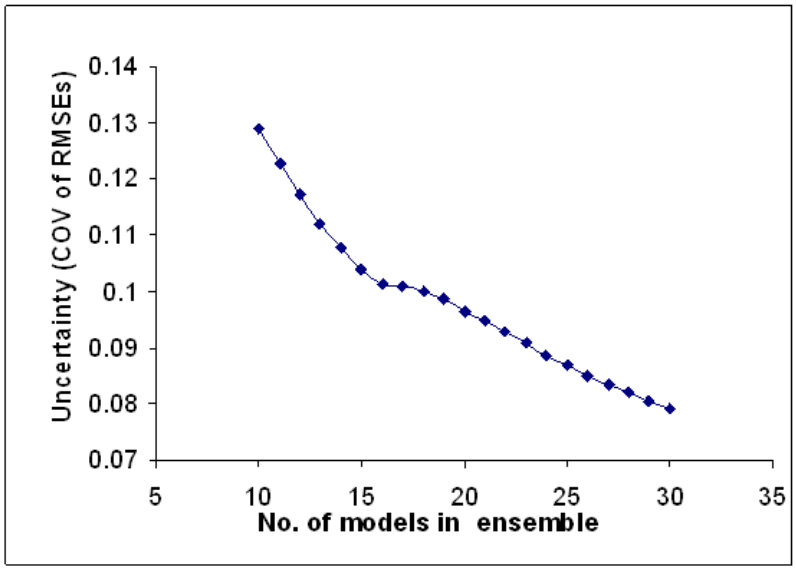

Figure 4.9 Uncertainty levels for increasing ensemble size for salinity C3

4.6.2 Multi-objective optimisation

The multi-objective optimisation algorithm NSGA-II was used to solve the optimisation formulations of both the multiple realisation and chance-constrained approaches. Similar to an ordinary genetic algorithm, NSGA-II has a population-based approach for deriving the optimal solutions. The population size used in this study is 200. NSGA-II was run for 750 generations to obtain the optimal solution. Thus a total of 200×750 evaluations of the aquifer response to specific groundwater extraction patterns would be required before obtaining the solutions. The NSGA-II parameters used were crossover probability – 0.9 and mutation probability – 0.02. The sensitivity of the optimal solution to population size, number of generations, and NSGA-II parameter values was evaluated by conducting a number of numerical experiments by running the NSGA-II model with different combinations of these parameters. It was found that for the number of generations less than 750 and population size less than 200, convergence to the Pareto-optimal front is not achieved. However, convergence is obtained for a smaller population size for a larger number of generations. It is noted that reducing the population size affects the spread of solutions in the Pareto-optimal front. Some regions of the Pareto-optimal front get eliminated as a result of reduction in the population size. The optimisation problems have 33 variables which are the pumping rates from 11 locations for 3 time periods. The optimisation by multiple realisation approach has 90 constraints, corresponding to 3 ensembles with 30 surrogate models each predicting the salinity levels C1, C2 and C3.

4.6.3 Pareto-optimal front

Pareto-optimal solutions refer to a non-dominated front of solutions obtained for the coastal aquifer management problem. On the Pareto-optimal front any improvement in one objective function requires a corresponding decline in the other objective function. These sets of solutions are obtained for the coastal aquifer management problem using multi-objective optimisation for both multiple realisation and chance-constrained approaches. All the solutions on the front are non-dominated and the water managers can choose a prescribed solution to implement a specific pumping pattern so as to maximise the benefits while simultaneously limiting the aquifer contamination. The Pareto-optimal solutions for different reliabilities obtained by the multiple realisation and chance-constrained methods are compared in figures 4.10 –

4.12. For the multiple realisation approach, the reliability refers to the per cent of surrogate models in the ensemble, the imposed constraints of which are satisfied in the optimisation. For the chance-constrained method, the reliability is obtained from the inverse cumulative distribution function of the residuals in the salinity prediction by the ensemble of surrogate models for salinities C1, C2 and C3.

Figure 4.10 illustrates the Pareto-optimal front for a reliability of 0.99. In the multiple realisation approach this set of solutions satisfies the constraints imposed by all the surrogate models linked with the optimisation model. In the chance-constrained formulation this set of solutions corresponds to an error in prediction, corresponding to a reliability of 0.99. Similarly, figures 4.11, 4.12 and 4.13 illustrate the Pareto-optimal fronts corresponding to reliability levels of 0.8, 0.66 and 0.5. Figure 4.13 also compares the fronts of reliability level 0.5 to the Pareto-optimal front obtained using single surrogate model in optimisation.

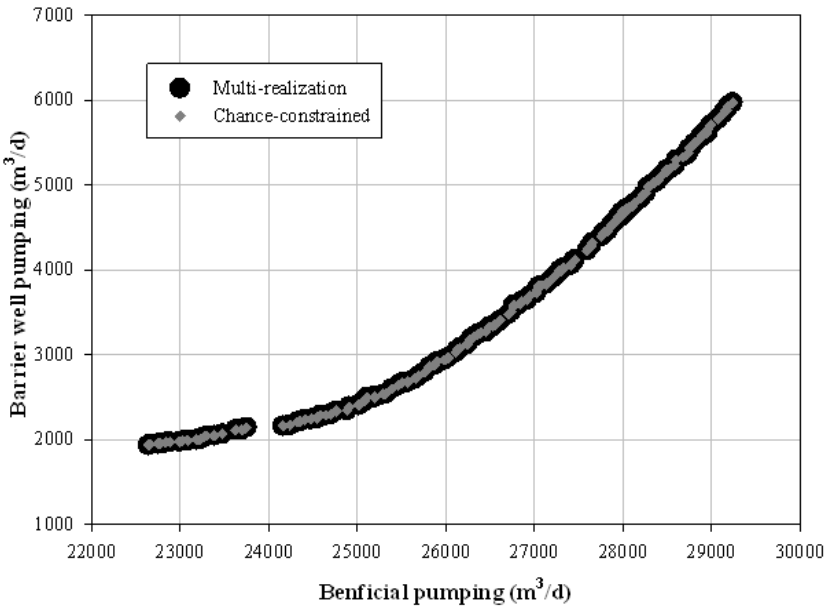

Figure 4.10 Pareto-optimal fronts with reliability 0.99

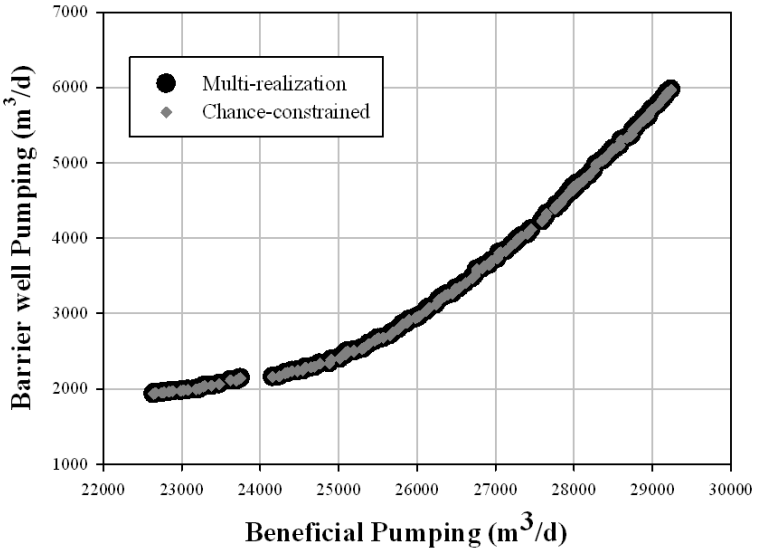

Figure 4.11 Pareto-optimal fronts with reliability 0.8

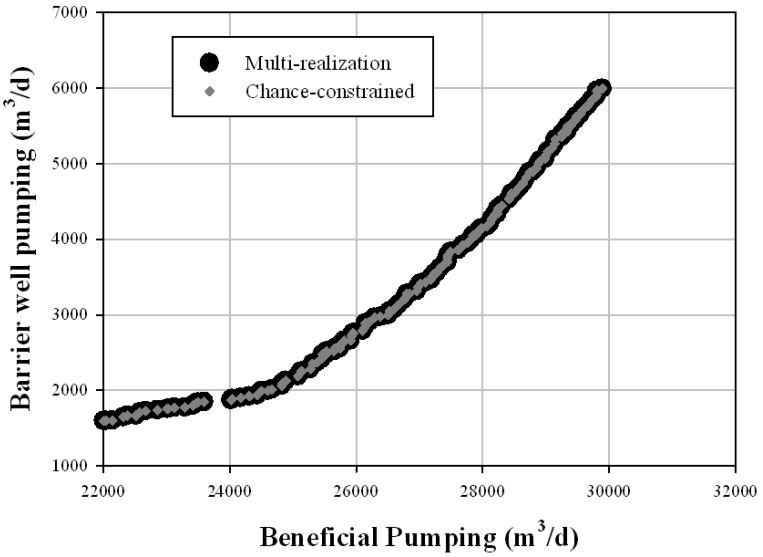

Figure 4.12 Pareto-optimal fronts with reliability 0.66

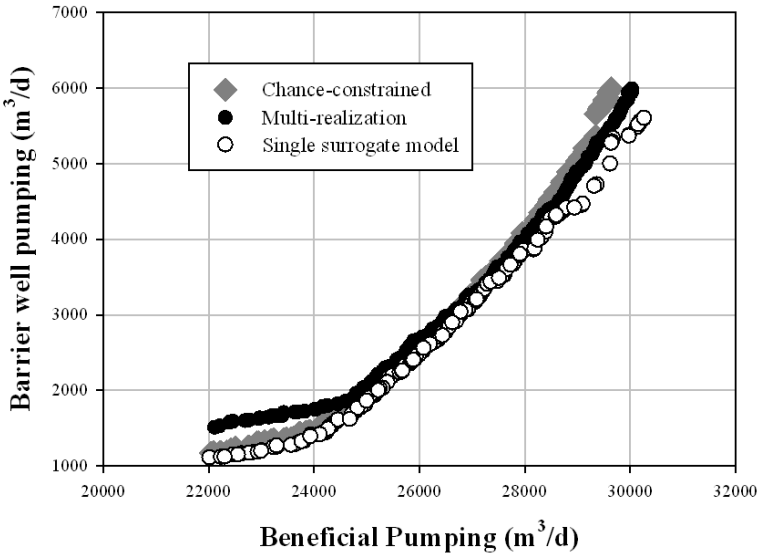

Figure 4.13 Pareto-optimal front for single surrogate model compared with fronts with reliability 0.5

The cumulative distribution functions (CDF) corresponding to C1, C2 and C3 are shown in figures 4.14, 4.15 and 4.16. The errors are more or less symmetrically distributed with a probability of 0.5 for zero residual in all the three cases.

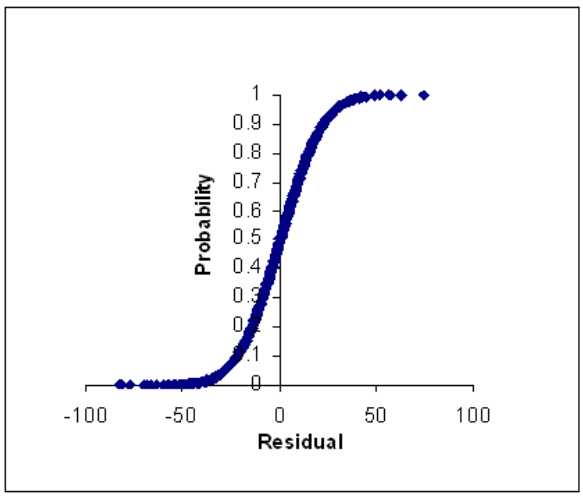

Figure 4.14 CDF for the residuals in the ensemble predictions of salinity C1

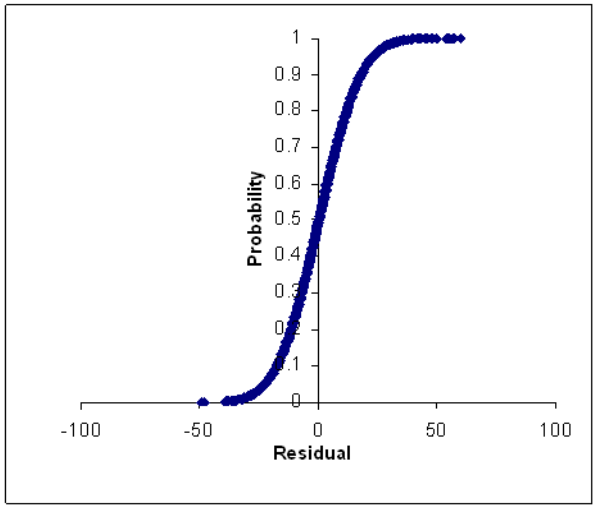

Figure 4.15 CDF for the residuals in the ensemble predictions of salinity C2

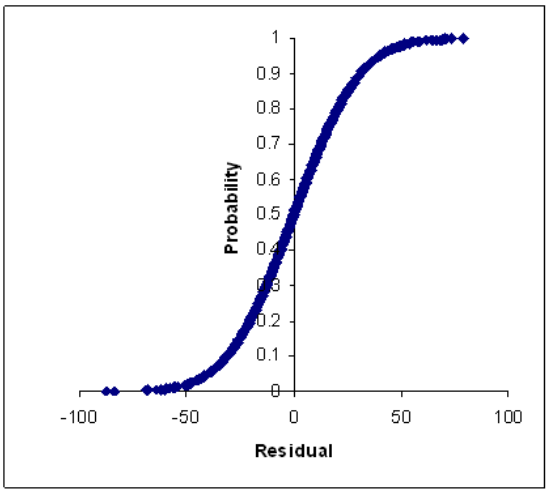

Figure 4.16 CDF for the residuals in the ensemble predictions of salinity C3

It can be noted that Pareto-optimal solutions with a higher reliability level appear to be inferior to those with a lower reliability level. The plausible reason is that, as reliability decreases the probability of these solutions violating the constraints increases. Therefore, the apparently better solutions may not be feasible. In figure 4.13, the Pareto-optimal front obtained for a reliability level of 0.5 is compared against the Pareto-optimal front obtained using only the best surrogate model in the coupled simulation-optimisation. It could be observed that the front obtained using the single surrogate model is very close to, and slightly better than the fronts obtained for a reliability level of 0.5 obtained using the multiple realisation and chance-constrained methods. In accordance with the general trend of variation of the Pareto-optimal front with the reliability, it could be deduced that the reliability level of the solutions obtained using a single surrogate model linked with the optimisation algorithm is less

than 0.5. In using a single best surrogate model in the coupled simulation-optimisation it is assumed that the surrogate model prediction has a 0 residual, i.e. the surrogate model simulation is equivalent to the numerical model simulation. However, it can be observed from the cumulative distribution functions that the probability of zero residual is 0.5. Since most of the optimal solutions are limit state designs, i.e. optimal solutions lie on the constraint bounds, the uncertainty in the surrogate model structure often causes the optimal solution to move into the infeasible region.

Salinity levels corresponding to five different optimal solutions in the Pareto-optimal front obtained using the best surrogate model in the coupled simulation optimisation model are shown in table 4.2. It could be observed that, in the optimal solutions, the salinity levels C1 and C3 converge to the permissible maximum concentration and hence the solutions are on the constraint boundaries. Hence, a small error in the surrogate model prediction can move these solutions into the infeasible zone. The salinity levels corresponding to these solutions are simulated using the actual simulation model and are compared with the values obtained using the surrogate model. It could be observed that some of the actual salinity levels obtained from the numerical simulation model violate the constraints, thus forcing the derived optimal solutions into the infeasible zone. The errors in the predicted salinity level for the optimal solutions are given in tables 4.3 and 4.4. Tables 4.3 and 4.4 correspond to multiple-realisation and chance-constrained approaches respectively. The errors refer to the difference in the salinity levels obtained using the actual numerical simulation model and the surrogate model. In both the cases, it is evident that the errors are less when the reliability level is high.

Table 4.2 Salinity levels corresponding to 5 optimal solutions from single surrogate model based optimisation

	C1 $\leq$ 0.5 kg/m^3		**C2 $\leq$ 0.6 kg/m^3**		**C3 $\leq$ 0.6 kg/m^3**	
Solution No.	**SM $\times 10^{-3}$ kg/m^3**	**NM $\times 10^{-3}$ kg/m^3**	**SM $\times 10^{-3}$ kg/m^3**	**NM $\times 10^{-3}$ kg/m^3**	**SM $\times 10^{-3}$ kg/m^3**	**NM $\times 10^{-3}$ kg/m^3**
1	500.00	483.04	563.39	561.45	599.99	622.05
2	500.00	515.33	583.13	575.97	599.99	623.52
3	500.00	510.34	582.39	573.16	599.99	599.76
4	500.00	483.00	574.68	548.73	599.99	624.23
5	500.00	498.25	574.48	563.57	599.99	618.35

SM = Surrogate model NM = Numerical model

Table 4.3 Residuals in salinity prediction for 5 optimal solution obtained by multiple realisation optimisation

Reliability	0.99			0.8			0.66			0.5		
Solution No.	C1	C2	C3	C1	C2	C3	C1	C2	C3	C1	C2	C3
1	3.16	0.61	-4.24	1.83	12.09	8.21	-9.89	1.77	-14.08	30.92	28.19	8.20
2	6.14	-0.29	5.21	-4.19	3.55	-4.12	17.09	-0.92	-21.90	-8.53	-19.43	-2.82
3	4.93	0.01	4.89	-6.00	12.66	12.57	-2.58	0.00	6.98	27.14	-22.50	6.96
4	-0.05	0.44	5.27	-4.52	-2.50	-12.39	-5.22	0.07	1.05	-18.52	-28.29	17.87
5	2.27	-0.36	-0.04	-5.60	7.70	6.02	8.76	0.39	5.73	-10.23	30.06	-9.14

Table 4.4 Residuals in salinity prediction for 5 optimal solution obtained by chance-constrained optimisation

Reliability	0.99			0.8			0.66			0.5		
Solution No.	C1	C2	C3	C1	C2	C3	C1	C2	C3	C1	C2	C3
1	-3.09	0.49	4.55	1.85	6.28	-1.64	9.64	-1.78	-13.98	-20.64	7.34	14.19
2	-1.13	0.27	-3.47	-6.64	9.14	-6.60	-12.30	0.89	-20.85	-27.10	-8.90	-0.63
3	2.97	0.24	-5.75	1.93	-6.43	4.79	-9.13	1.99	-12.67	21.80	30.68	-23.04
4	-0.62	-0.46	5.10	-2.79	-0.60	-10.08	6.86	2.11	-3.32	-12.16	-27.14	-26.31
5	-3.44	-0.54	-0.76	0.78	6.06	-9.87	-22.46	1.09	-17.28	28.64	-16.12	5.16

The ensemble-based surrogate modelling approach quantifies the uncertainties in the model structure and parameters. Reliable optimal solutions for coastal aquifer management were obtained using the ensemble surrogate models with the stochastic multiple realisation and chance-constrained optimisation models.

4.6.4 Sensitivity analysis

Comparison of Pareto-optimal fronts for different reliabilities show that for 30 surrogate models in the ensemble, the multiple realisation approach identifies the same front as the chance-constrained optimisation approach for identical reliability levels. This implies that the constraints imposed by stochastic optimisation using multiple realisation are as rigid as the chance-constraints when the number of surrogate models in the ensemble is large enough to quantify the uncertainty in the model structures and parameters.

In order to investigate the effect of the number of surrogate models in the ensemble on the multiple realisation optimisation approach for each reliability level, numerical experiments were performed with 15, 10 and 5 models in the ensemble. The corresponding Pareto-optimal fronts for reliability level 0.99 are compared with the fronts obtained using 30 models and the chance-constrained model is shown in figure 4.17. As the size of the ensemble decreases, the fronts move further to find

seemingly better solutions, which actually may be infeasible solutions. Similar results were obtained for other reliability levels. Hence it can be inferred that the size of the ensemble has an effect on the stochastic optimisation using multiple realisations.

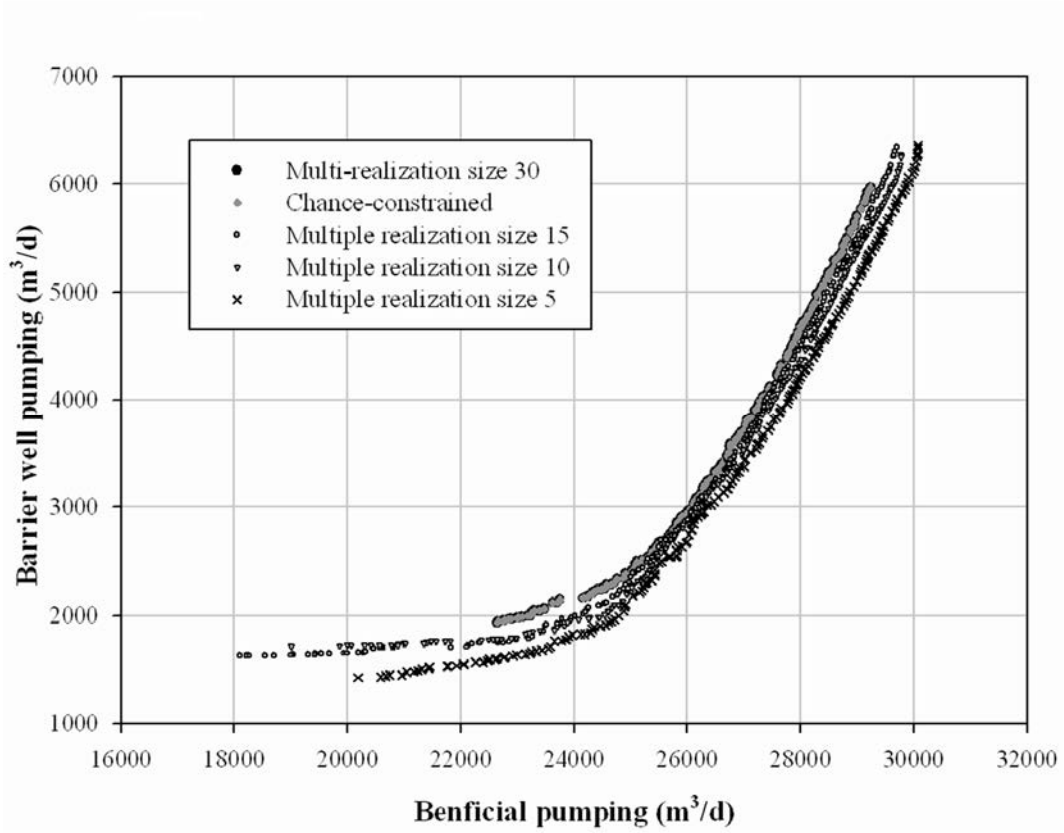

Figure 4.17 Sensitivity of the solutions to the ensemble size

4.7 Summary and Conclusions

Surrogate models are useful tools as substitutes for complex numerical simulation models in coupled simulation-optimisation approaches for solving groundwater management problems. However, their practical applications have been limited primarily due to the reliability of the surrogate model predictions. The reliability of surrogate model predictions is dependent on the uncertainties in the model structure and parameters. The uncertain surrogate models when used in a coupled simulation-optimisation framework affects the quality as well as the reliability of the optimal solutions obtained. Since most optimal design solutions are limit state in nature the error in the surrogate model predictions could make the derived optimal solutions even infeasible. In order to address these issues and as a possible remedy, this study proposed and evaluated the performance of an ensemble of surrogate models based simulation-optimisation model. The ensemble of surrogate models is also used to

quantify the uncertainty in the surrogate model structure and parameters. Salinity prediction by each surrogate model in the ensemble differs from others due to the model structure and parameter uncertainty. Two different optimisation formulations were used to derive the optimal abstraction rates. In the first method, each surrogate model in the ensemble was independently linked to the multi-objective genetic algorithm NSGA-II, using the multiple realisation formulation. In the second method, the errors in salinity predictions were quantified using the ensemble of models and the cumulative distribution function of the errors was obtained. Based on the cumulative distribution function, the chance-constrained optimisation problem was formulated and solved using the multi-objective genetic algorithm NSGA-II. The reliability of the chance-constrained model is analogous to the reliability obtained using the ensemble surrogate model approach, as the management model is constrained by the permissible maximum limits on salinity concentrations. The Pareto-optimal sets of solutions obtained using the two methods for different reliability levels were compared. These fronts were also compared with the Pareto-optimal set obtained using the best surrogate model in the coupled simulation optimisation. It was observed that the front obtained using the single surrogate model in the optimisation was close to the front corresponding to a specified reliability of 0.5. It could be argued that the reliability of the optimal solution obtained using a single surrogate model in the linked simulation optimisation model for coastal aquifer management roughly corresponds to 0.5. However, using an ensemble of surrogate models with stochastic optimisation helps improve the reliability of the salinity predictions and subsequent optimal solutions.

Ensemble-based surrogate modelling in couple-simulation optimisation has significant advantages over the single surrogate modelling approach. The single surrogate modelling approach does not take into consideration the predictive uncertainty and assumes that the surrogate model prediction is equivalent to numerical simulation. The ensemble-based methodology is able to quantify the predictive uncertainty and use it in a stochastic optimisation model. Thus the ensemble-based approach accounts for the error in surrogate model prediction due to predictive uncertainty which is difficult to accomplish using a single surrogate model. The ensemble-based approach is found to derive more reliable optimal solutions while retaining the computational advantages of surrogate modelling approach.

It should be possible to use ensemble surrogate models in coupled simulation-optimisation groundwater management studies considering the uncertainty in the groundwater parameters. An ensemble of surrogate models could be used to substitute groundwater models with different hydraulic conductivities and other uncertain parameters. For this, each member of the ensemble has to be trained using a different data set obtained by using a particular realisation of the uncertain groundwater parameters in the numerical simulation model. The ensemble can be then used in a stochastic optimisation framework to derive groundwater management strategies under groundwater parameter uncertainty. This is explored in chapter 5.

5. STOCHASTIC AND ROBUST OPTIMAL PUMPING STRATEGIES FOR COASTAL AQUIFER MANAGEMENT AND APPLICATION TO LOWER BURDEKIN COASTAL WELL FIELD

A similar version of this chapter has been submitted to *Journal of Hydrology*.

5.0 Overview

The methodologies so far developed in this work assumed deterministic values of groundwater parameters like hydraulic conductivity and groundwater recharge in the simulation model to derive optimal pumping strategies for coastal aquifer management. However, the groundwater flow and transport system itself being characterised by uncertain parameters, using a deterministic surrogate model to substitute it is an unrealistic approximation of the system. To date, few studies have considered stochastic surrogate modelling to develop groundwater management methodologies. In this chapter, a new methodology for coastal aquifer management under parameter uncertainty is explored. Genetic programming based ensemble surrogate models are extended to characterise coastal aquifer responses to pumping under parameter uncertainty. These approximation surrogates are then coupled with multiple realisation optimisation for the stochastic optimisation of groundwater management in coastal aquifers. The methodology retains the computational efficiency of the surrogate modelling approach and at the same time address the parameter uncertainty in groundwater modelling in the purview of groundwater management. Uncertainties in hydraulic conductivity and the annual aquifer recharge are incorporated.

5.1 Stochastic and robust optimal pumping strategies for coastal aquifers

This chapter describes the development of a methodology for the stochastic and robust optimal management of pumping in coastal aquifers. The same objectives of optimisation as previously used, i.e. maximisation of total beneficial pumping and minimisation of total barrier pumping are considered. The methodology addresses uncertainty in the values of hydraulic conductivity and groundwater recharge. The ensemble surrogate modelling approach together with multiple-realisation

optimisation proposed in chapter 4 is extended for stochastic optimisation under parameter uncertainty. The same multi-objective optimal management considering beneficial and barrier well pumping is considered. The mathematical formulation of the problem considering parameter uncertainty is given as follows:

$$Maximize \sum_{p \in PROD} \sum_{t \in T} Q_t^p \tag{5.1}$$

$$Minimize \sum_{b \in BAR} \sum_{t \in T} q_t^b \tag{5.2}$$

$$c_i = f_i(Q_t^p, q_t^b, \boldsymbol{\theta}) \tag{5.3}$$

$$c_i \leq c_i^{\max} \tag{5.4}$$

$$Q_{\min} \leq Q_t^p \leq Q_{\max} \tag{5.5}$$

$$q_{\min} \leq q_t^b \leq q_{\max} \tag{5.6}$$

where, Q_t^p is the pumping from the p^{th} beneficial pumping well for the t^{th} time period, q_t^b is the pumping from the b^{th} barrier well for the t^{th} time period and c_i is the salinity at the monitoring location i at the end of the management time frame considered in the optimisation model, resulting from the pumping. *PROD* and *BAR* designate respectively, the set of all production wells and barrier wells in the well field. c_i is a function of Q_t^p and q_t^b and also the numerical model parameter set $\boldsymbol{\theta}$. The function f_i represents the numerical simulation model. When the parameter set $\boldsymbol{\theta}$ is considered as stochastic, solving this optimisation model would imply testing each solution comprising of a set of pumping values against multiple realisations of the uncertain parameter set $\boldsymbol{\theta}$. $Q_{\min}$ and $Q_{\max}$ are respectively the lower and upper limits on the production well pumping, and $q_{\min}$ and $q_{\max}$ are the corresponding values for the barrier well pumping.

Due to the computational difficulties in implementing this optimisation scheme, the ensemble surrogate model is used as an approximate substitute of the simulation model f_i within the optimisation model. The ensemble surrogate model based multiple realisation optimisation implements the reliability concept in the following manner.

$$c_i^r \approx \zeta_i^r(Q_t^p, q_t^b, U) \tag{5.7}$$

$$U = \psi(\boldsymbol{\theta}, \boldsymbol{\omega}) \tag{5.8}$$

$$\beta = r/R \tag{5.9}$$

$$c_i^r \leq c_i^{\max} \quad \forall r \text{ such that } \sum r \geq R\beta \tag{5.10}$$

The concentration c_i is approximated using c_i^r, which are *r* different values of concentration at the i^{th} monitoring location, obtained from different surrogate models in the ensemble. The functional relationship between the pumping and the resulting salinity level is approximated by *r* realisations of the salinity obtained from different surrogate models given by ζ_i^r for each location *i*. The realisations c_i^r are different from each other because of the uncertainty *U* in the surrogate models, which is a function ψ of both numerical model parameters, $\boldsymbol{\theta}$ and the surrogate model structure and parameters, $\boldsymbol{\omega}$. Reliability is defined in 5.9 as the ratio of number of realisations *r* which satisfies the constraint on the limit of concentration $c_i^{\max}$ to *R*, the total number of realisations of salinity obtained from the ensemble prediction models. Constraint (5.10) ensures that all realisations c_i^r which belongs to a set of realisations with a size of at least $r = R\beta$ should satisfy the limit on the concentration given by $c_i^{\max}$. Thus, the Pareto-optimal front for the multi-objective management problem is derived for a specific reliability level β, which is chosen by the manager depending on how reliable the solutions need to be.

Another variant of the simulation optimisation formulation was developed which utilises only a single surrogate model trained and tested using a bootstrap sample having a higher size than the original data set. The objective is to investigate the possibility of a single surrogate model which can predict the saltwater intrusion process with reasonable accuracy so that optimal solutions for aquifer management may be derived. The mathematical formulation of this coupled simulation-optimisation problem is similar to the previous one except that the equations (5.7)–(5.10) are replaced by a single equation given as follows:

$$c_i = \xi(Q_t^p, q_t^b) \tag{5.11}$$

5.2 Ensemble surrogate modelling approach

The coastal aquifer response to pumping in terms of the salinity levels at specified monitoring locations is approximated using genetic programming based surrogates. Input-output patterns of pumping and resulting salinity levels obtained from the 3D coupled flow and transport simulation model FEMWATER is used to train the surrogates. Hydraulic conductivity and aquifer recharge were considered as uncertain parameters in the model development. An ensemble of surrogate models was used to implicitly account for the uncertainty in the salinity prediction due to the parameter uncertainty in the model development. The detailed methodology of the ensemble surrogate model development is as follows.

5.2.1 Parameter uncertainty characterisation and training set generation

The developed framework for optimal pumping management from coastal aquifers considers the uncertainty in the hydraulic conductivity and the recharge from rainfall occurring over the time horizon of operation. In the present study, hydraulic conductivity is assumed to be homogeneous but uncertain within a range which is obtained from the values measured in the field. To generate a representative set of hydraulic conductivity realisations, Latin Hypercube Sampling was performed on a log-normal distribution of hydraulic conductivity. The prescribed distribution that represents the uncertainty in the hydraulic conductivity value was divided into N equi-probable intervals. A single value was selected randomly from each interval.

Similarly Latin Hypercube Samples of normally distributed aquifer recharge values were generated to constitute a representative set of probable aquifer recharges. The hydraulic conductivity values and aquifer recharge values are then randomly paired to constitute the random realisations of the uncertain parameter values.

Uniformly distributed Latin Hypercube Samples of the pumping variables were generated from the variable space bounded by the minimum and maximum rates of pumping possible from the considered pumping locations. One set of values of pumping from the well locations together with the corresponding aquifer response in terms of the salinity levels at the monitoring locations, obtained corresponding to a specific set of the uncertain parameters, form a single pattern of pumping. To generate each input-output pattern, the pumping inputs and the values of the uncertain parameters are chosen at random. Different pumping patterns are input into the

simulation model with a random choice of the parameter values to compute the concentration outputs for this combination of pumping and parameter values.

The salinity levels at the monitoring locations were simulated using the 3D density-dependent flow and transport model FEMWATER. Different uncertain parameter sets and different pumping patterns generated as described above were used in repeated runs of the simulation model to simulate the aquifer processes in the coastal aquifer. This resulted in the generation of a data pool of pumping-salinity patterns corresponding to different realisation of the uncertain parameters.

5.2.2 Bootstrap sampling

A non-parametric bootstrap method was used to generate different realisations of the original data pool of the pumping-salinity patterns. The key idea is to generate multiple realisations of the data set having different representations of the pumping decision space and uncertain parameters. Bootstrap samples are generated by repeated sampling with replacement from the original data set. This method has been previously used in developing ensemble models by Parasuraman and Elshorbagy (2008) and Sreekanth and Datta (2011).

The original data pool was divided into two sets called the original training set (TR) and original testing set (TE) each having a size N. A bootstrap size B is specified so that B different sample sets each of the original training (TR_B) and testing sets (TE_B) are generated after the sampling procedure. Random sampling with replacement from the original training set and testing set was performed to generate the bootstrap sample set. Thus the patterns in the bootstrap sample sets are a subset of the patterns in the original data sets. However, some patterns are absent and some patterns are repeated with different frequencies in the bootstrap sample sets.

5.2.3 Surrogates

In the present work, an ensemble of surrogates are developed using genetic programming adopting a methodology similar to the one described in chapter 4. Each surrogate in the ensemble is trained and tested using a bootstrap sample set generated from the original training and testing data set. When these bootstrap samples, which contain repeated samples of pumping-salinity patterns, are used in the training, different weightings of patterns occur in the objective function used in the GP to develop the surrogate. Since the pumping-salinity patterns in the data pool correspond to different combination of uncertain parameters, it results in the development of

surrogates which are differently capable of making predictions in different regions of the decision-parameter space. Also, since the pumping patterns are repeated in the training and testing data the resulting surrogates are different in their capability to make predictions in different regions of the pumping variable space.

The methodology of developing ensemble surrogates is illustrated in figure 5.1. With sufficient representation of the entire parameter and decision space in the original data set and sufficient number of surrogates in the ensemble, the ensemble surrogate modelling approach can achieve sufficiently accurate approximation of the saltwater intrusion prediction at the selected monitoring locations. Sufficient accuracy of prediction is verified by validating the optimal solutions by checking the corresponding salinity levels at the monitoring locations using the numerical simulation model. If the prediction errors are low and the optimal solutions are still in the feasible domain, it may be considered that the representation of the parameter-decision space is sufficient. Else, an adaptive training approach, as proposed by Yan and Minsker (2006) or Sreekanth and Datta (2010), may be required to improve the surrogate models. This may be required for more complex applications.

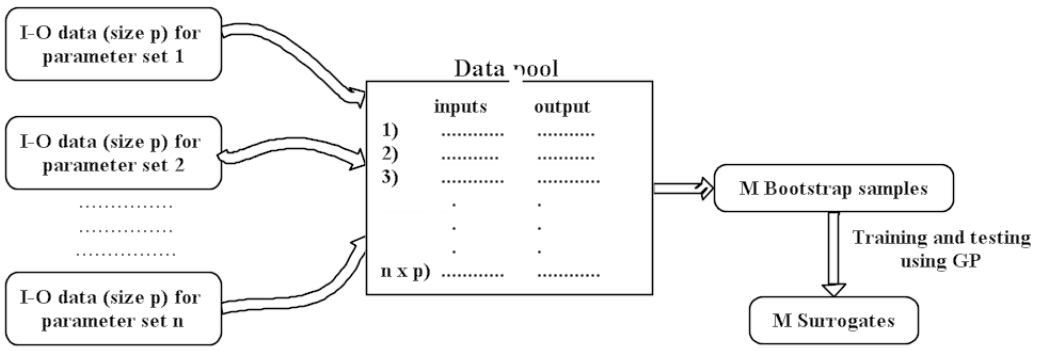

Figure 5.1 Schematic representation of ensemble surrogate model development

5.2.4 Ensemble size and the uncertainty in the prediction of the ensemble surrogates

The number of surrogate models in the ensemble is determined by a criterion based on the uncertainty in the ensemble prediction. The method was implemented as follows. Root mean square errors of the individual surrogates in the ensemble were computed. Overall performance of the ensemble modelling approach was quantified by calculating the mean and standard deviation of the root mean square errors. An initial ensemble size with a few surrogates was chosen. The ensemble performance was

determined using the coefficient of variation (CoV) of the RMSEs. CoV was calculated as standard deviation over the mean value. Then surrogates were added one by one and the CoV was recalculated. The ensemble size for which the CoV achieves a minimum value, when the difference between two consecutive CoV values are less than a limiting value, was selected as the actual ensemble size for the linked simulation optimisation.

The predictive uncertainty of the ensemble surrogates may be attributed to two causes. The parameter uncertainty is represented by Latin Hypercube Samples of parameters in the numerical model development. This uncertainty is reflected in the pumping-salinity patterns generated using the flow and transport model. This parameter uncertainty propagates into the predictions through the surrogate models. The second cause of predictive uncertainty is the uncertainty inherent in the surrogate model structure and parameters as discussed in chapter 5 and Sreekanth and Datta (2011).

5.3 Coupled simulation-optimisation

Multiple realisation optimisation (Wagner & Gorelick 1987, 1989; Morgan, Eheart & Valocchi 1993; Chan 1993; Feyen & Gorelick 2005; Bayer, Buerger & Finkel 2008) together with an ensemble surrogate modelling approach as proposed in chapter 4 is used to solve the coastal aquifer pumping management problem considering uncertainty in the parameters. Here, the multiple realisations refer to the salinity values obtained from different surrogate models in the ensemble. The surrogates developed using genetic programming are coupled to the multi-objective genetic algorithm NSGA-II (Deb 2001) used in this study. The methodology incorporating the parameter uncertainty is schematically represented in figure 5.2.

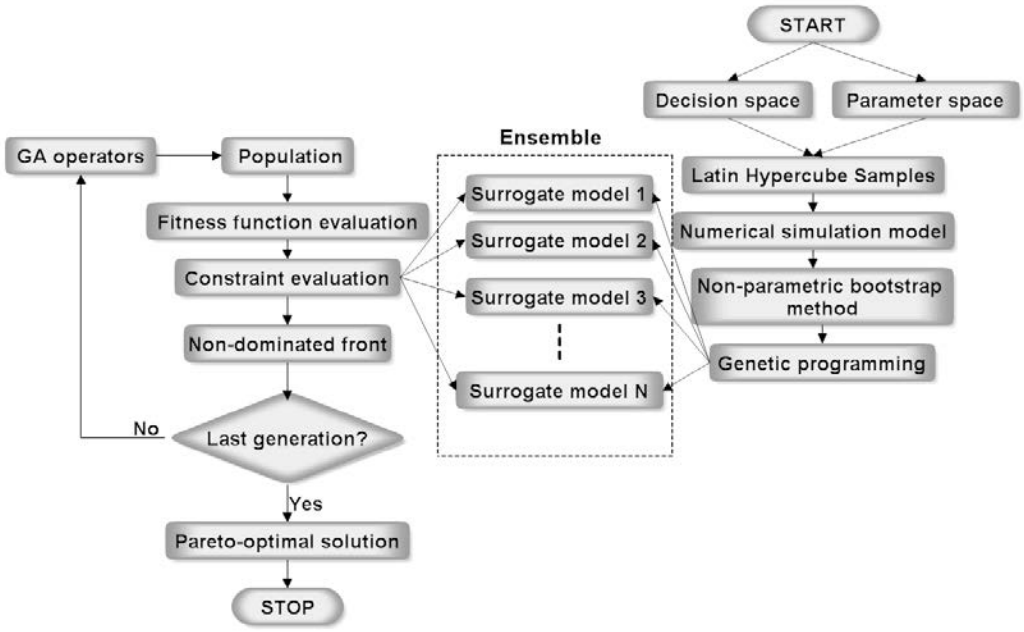

Figure 5.2 Schematic representation of the ensemble based simulation-optimisation

In the coupled simulation optimisation methodology all the surrogates are coupled with the multi-objective genetic algorithm NSGA-II in such a way that during each iteration of the optimisation search, all the surrogates are called upon for predicting the approximate value of the concentration. Thus, there are as many predictions of the concentration at a location, as the number of surrogates in the ensemble. Thus, multiple realisations of the concentration value are obtained from the ensemble of surrogates. The optimisation algorithm searches for the optimal pumping strategy which limits the concentration at the monitoring locations to prescribed values. In this multiple realisation approach, reliability is implemented as the percentage of the surrogates in the entire ensemble with concentration predictions that do not violate the imposed constraints of maximum concentration levels (Feyen & Gorelick 2005; Sreekanth & Datta 2011).

5.4 Performance evaluation

The performance of the methodology is evaluated by applying it to a well field in a coastal aquifer system in the Lower Burdekin in Queensland, Australia. Development of the conceptual model for simulating density-dependent flow and transport for the well field is described in this section.

5.4.1 Geography and geological characterisation

The Burdekin is a deltaic region in North Queensland extending over 850 km^2 in area and the primary land use is agriculture with extensive areas falling under sugarcane cultivation (McMahon et al. 2000). About 40000 hectares of area is irrigated for cultivating sugarcane (Narayan, Schleeberger & Bristow 2007). For more than 120 years, the sugarcane industry has been active and groundwater has been extensively used for irrigation, although since the construction of the Burdekin Falls Dam, a considerable increase in the use of surface water has occurred. Two autonomous water boards regulate the use of water in this area. Groundwater augmentations using artificial recharge have been in place since the 1960s.

Figure 5.3 Rita Island study area (Google Earth Pro, 2010)

In this work, Rita Island, in the Burdekin Delta, with an area of 60 km^2 was chosen for the performance evaluation study (Figure 5.3). The area is essentially flat and is bounded by distributaries of the Burdekin River on two sides. On the eastern boundary of the study area is the Coral Sea. Part of the area is irrigated for sugarcane cultivation and the rest is an uncultivated area towards the sea. Annual average rainfall of the region is 1032 mm, the majority of which occurs in the wet season between the months of December to March. Annual average evaporation is 2062 mm. Groundwater exists in an unconfined aquifer with an average depth of 60 metres overlying bedrock of granitic origin. The aquifer material mainly consists of sediments deposited by the river over a long period of time. It comprises of a

combination of clean sand with distributions of thin layers and lenses of low permeable clay. The sediment layers often lack lateral continuity with different layers inter-fingering into each other in a zigzag fashion. Due to this zigzag arrangement of low permeable clay layers in between the high permeable sand, it is estimated that on a regional scale, there exists hydraulic continuity between the sandy layers (McMahon et al. 2000).

5.4.2 Regional scale simulation model

A regional scale groundwater flow and transport model was developed for the specific study area using the FEMWATER model. The study area with the boundaries and well locations are shown in figure 5.4. This specific area was chosen because of the presence of well defined boundary conditions in terms of the river reaches on two sides and the sea on the third side. This would permit the use of known constant head boundary conditions for all the three boundaries of the study area. Precise knowledge of the boundary conditions helps in achieving realistic quantification of the saltwater intrusion rates. In this work, available data on the groundwater parameters and average boundary conditions are used to develop realistic models to simulate the pumping-induced saltwater intrusion. From the previously published data (McMahon et al. 2000, Narayan, Schleeberger & Bristow 2007), there exists considerable uncertainty in hydraulic conductivity values for the area. This is accounted for in the developed model by using Latin Hypercube samples from the log-normal distribution of hydraulic conductivity values obtained for the area. A similar approach is used to characterise the unknown annual rates of groundwater recharge into the aquifer.

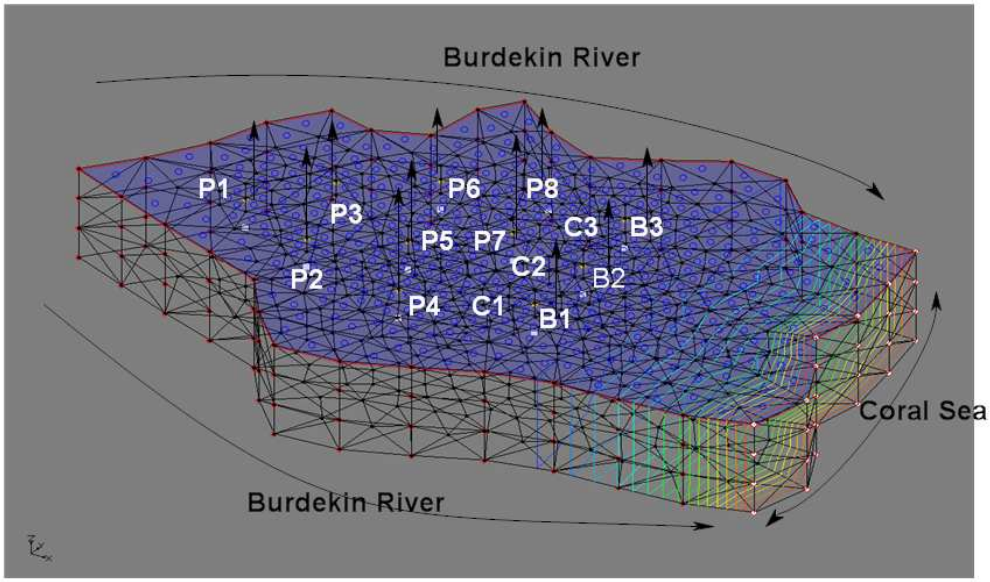

Figure 5.4 3D view of the well field with well locations, control points and boundaries

5.4.3 Modelling pumping-induced saltwater intrusion

The data obtained from different sources and previous published work was used in the model development. As this part of the Burdekin delta has not implemented regulated pumping, accurate data on groundwater pumping was not available. However, a range of values of pumping have been estimated based on previous studies (McMahon et al. 2000).

5.4.4 Conceptual model

A conceptual model of the Burdekin study area was developed using the geographical and other input data collected from different sources. A map of the area was geo-referenced using the geographical co-ordinates of three corners of the study area. Then, all the collected data were overlaid on the map as different ‘coverages’, where ‘coverage’ designates ‘layers’ in GIS terms. These include the boundary conditions, well locations and pumping rates, initial hydraulic head, initial concentrations and saltwater wedge location.

5.4.5 Boundary conditions

An unconfined aquifer with a single layer of sediment materials with an average depth of 60 metres is considered for modelling groundwater flow and transport. The aquifer overlies bedrock of granitic origin. The considered area has three boundaries as shown in figure 5.4. The Burdekin River bifurcates at the western corner of the study area. The two channels of the river flow along the two boundaries on the north and

south of the modelled area. Dirichlet boundary conditions are assumed along these two boundaries with two different constant head values for two main seasons of the year. These values correspond to the average head of water maintained in the river in the wet and dry seasons. The seaside boundary on the east was assigned an average constant head value of 0 metres corresponding to the mean sea level. A constant concentration value of 35 kg/m^3 was assigned as the concentration boundary condition for the seaward boundary. Zero concentration values were assigned for the river boundaries.

5.4.6 Pumping Wells

A large number of pumping wells are located within the considered study area. The well locations were obtained from the database of the Queensland Department of Environment and Resource Management. The wells are located within the conceptual model using their Universal Transverse Mercator (UTM) co-ordinates. Once the wells are assigned, the study area is discretised into finite elements in such a way that all the wells are located at the nodes in the finite element mesh. For developing the regional scale model, all the wells falling within a radial distance of 1 kilometre were assigned at a single node.

The exact rates of pumping from different well locations in Rita Island are unknown because the pumping in this area is not gauged or regulated. In order to have a realistic understanding of the possible range of saltwater intrusion as a result of pumping, 250 different pumping scenarios were considered. The total pumping from the Burdekin area is estimated to be 440–830 Mm^3/yr (McMahon et al. 2000). Based on the spatially averaged value obtained from this, the minimum and maximum possible pumping from each pumping node was determined as 0 and 13000m^3/d respectively. In the regional scale management model, one pumping node represents a group of wells located within a radial distance of 1 kilometre. The maximum pumping possible from that node is the sum of the maximum pumping rates for all the wells falling within this radial distance. With 10 to 15 wells falling within this radial distance the average draft from a single pumping well will be 865 m^3/d. Uniformly distributed patterns of pumping in this range were generated using Latin Hypercube Sampling for the considered beneficial and barrier well locations.

5.4.7 Initial conditions of hydraulic head and saltwater wedge location

The primary objective of developing the coupled flow and transport simulation model for the Burdekin study area is to study the saltwater intrusion phenomenon and possible hydraulic control of saltwater intrusion using barrier wells in coastal aquifers. A long-term transient simulation with average boundary conditions and stresses was performed to determine near steady state conditions that can be used as the initial conditions for the model. As pumping from the region has been active for almost 120 years, aquifer simulation was carried out for 120 years using the FEMWATER model. An average value of hydraulic conductivity of 32.67 m/d and a uniform recharge rate of 0.000484 m/d were specified. The initial conditions for this model assumed uniform groundwater heads and zero saltwater intrusion into the aquifer. At the end of 100 years of simulation, the near steady state hydraulic head values and the location of saltwater wedge, were obtained from the transient simulation results. These values compared satisfactorily with the limited amount of hydraulic head and concentration data available for the region, although no formal validation of the model was possible. These values were then used as the initial conditions for further models developed, considering it as a representative model for simulating the saltwater intrusion process for this region.

5.4.8 Three-dimensional finite element grid

After the development of the conceptual model, finite element mesh is generated over the study area, in such a way that all the wells in the study area are located at certain nodes of the mesh. The average size of the finite elements was chosen as 450 metres. But smaller element sizes of the order of 50 metres were used near the pumping wells. This is because pumping induces steep gradients of heads and concentrations at these locations and numerical stability may not be achieved if large elements are used for these regions. The aquifer space was vertically discretised into four layers. The pumping rate assigned to each well was equally distributed among the nodes falling between the well screens. Based on the initial condition obtained from the 120 years simulation, corresponding total head and concentration values are assigned to each node in the three-dimensional finite element grid.

5.4.9 Hydraulic conductivity and annual aquifer recharge

Hydraulic conductivity and annual aquifer recharge are considered as uncertain in the model development. Different estimates of these values have been reported in

previous studies. A wide range of hydraulic conductivity estimates have been reported for the Lower Burdekin area (McMahon et al. 2000; Narayan, Schleeberger & Bristow 2007). Narayan, Schleeberger and Bristow (2007) tested three different homogeneous hydraulic conductivity values, 10, 50 and 100 m/d in their saltwater intrusion simulation study using the two-dimensional SUTRA model. Homogeneous but uncertain values of the hydraulic conductivity are considered in this study. For this, a log-normal distribution of hydraulic conductivity with μ = log (32.67) and σ = 0.128 was used. Twenty-five Latin Hypercube Samples were selected from this distribution.

Aquifer recharges in the range 0.00136–0.00342 m/d are estimated (McMahon et al. 2000; Narayan et al. 2007) for the whole of the Lower Burdekin irrigation area. These high values are attributed to the groundwater recharge from the irrigation and artificial recharge schemes implemented in the area. The area used in this study is only partially irrigated and hence a smaller mean recharge value of 0.000484 m/d is used. The standard deviation value used is 0.000115. Latin Hypercube Samples were chosen from this normal distribution. These values were then paired randomly with the values of hydraulic conductivity to generate the random parameter space used in the simulation. These values are plotted in figure 5.5.

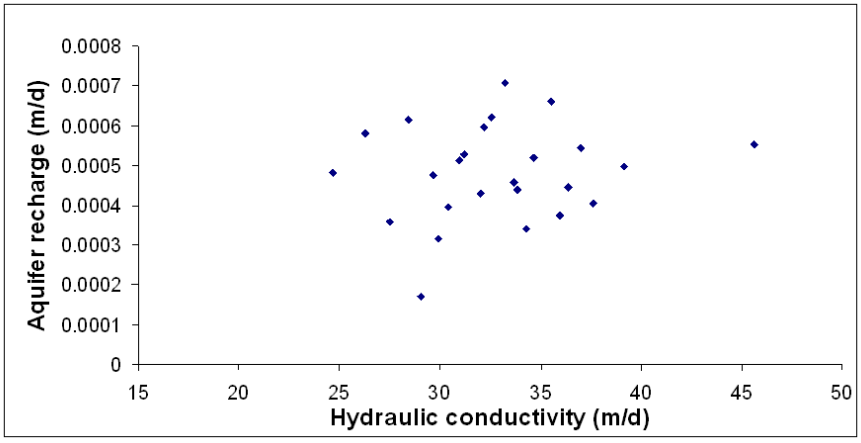

Figure 5.5 Latin Hypercube Samples of uncertain parameters

The three-dimensional coupled flow and transport simulation model FEMWATER was used to simulate the aquifer processes in response to pumping. FEMWATER uses a finite element method to solve the corresponding flow and transport equations. There is no barrier pumping well existing in this field. However, in this study we consider barrier wells pumping from the saltwater wedge as an

additional measure for hydraulic control of saltwater intrusion. Altogether, 8 pumping well locations and 3 barrier well locations were considered within the study area. Uniform extraction rates are assumed at each location for a time-step of one year. Other parameters used in the FEMWATER simulation model are as given in table 5.1.

Table 5.1 Simulation model characteristics

Characteristic	Value
Average aquifer thickness	60m
Longitudinal dispersivity	80m/d
Lateral dispersivity	35m/d
Molecular diffusion coefficient	0.69 m2/d
Soil porosity	0.2
Density reference ratio	7.14 x 10 -7
Mean hydraulic conductivity	32.67 m/d
Mean aquifer recharge	0.000484 m/d
Mean Y = ln K	1.51
Number of elements	4581
Number of nodes	1092

The aquifer responses were monitored at three nodes, at a depth of 19.8 metres, in between the pumping and barrier wells. The simulated concentration values at these nodes are named C1, C2 and C3 respectively as indicated in figure 5.4. The pumping management model prescribes permissible maximum concentration levels at these locations as 0.5, 0.6 and 0.6 kg/m^3 respectively. Corresponding to each combination of the uncertain parameters, 10 different pumping patterns were used to simulate the aquifer responses. Thus, 250 patterns of pumping were generated using Latin Hypercube Sampling and the FEMWATER model was used to simulate the aquifer processes to compute the concentration levels at the monitoring locations. Multiple training and testing sets were generated by using bootstrap sampling from the original data set of pumping-salinity patterns. For this, the original data set was split into two sets and bootstrap sampling was performed on each half to obtain separate training and testing sample sets. This ensures that training and testing of the surrogates are performed on mutually exclusive data sets.

The original training and testing data set had 150 and 100 patterns respectively. The respective bootstrap sample sizes used were 450 and 300. Thus each bootstrap sample set has thrice the number of patterns contained in the original data sets. This ensures that different patterns are repeated multiple times in different bootstrap data sets. However, increasing the bootstrap sample size largely may result

in identical sample sets which are undesirable. A pair of training and testing sample sets is used to develop one surrogate model. Similarly, different pairs of training and testing sample sets are used with genetic programming to generate the ensemble of surrogates.

These data sets were input to the genetic programming software Discipulus to generate the surrogate models. Forty models were required to reduce the uncertainty as described in section 5.2.4. C language based code of the developed models was generated and coupled with the NSGA-II algorithm for optimisation. For developing all the models, the parameters used in GP were as follows. The population size was specified as 500, mutation frequency 90, and crossover frequency 60. Optimum values of these parameters were chosen by trial and error with different parameters tested against the criteria of minimising the objective function of GP model training. The number of constants (the surrogate model parameters) used was limited to 30 in all the GP models These surrogate model parameters are comparable to the connection weights in a neural network surrogate model, in that way, the number of parameters used in the GP approach is very much less as compared to neural networks [*chapter 3*]. It was ensured that training and testing errors were in the same range to avoid over-fitting. Bloating was prevented in GP trees by dynamically adjusting the maximum size of the programs.

These surrogate models were coupled individually to the multi-objective genetic algorithm, NSGA-II (Deb 2001). For each generation, the new pumping solutions generated by the GA operators were tested using these surrogates for their constraint values of concentrations at the monitoring locations as described in the formulation. Another bootstrap sample of training and testing sample set of a larger size was generated from the original data set to investigate the possibility of a single universal surrogate. The sizes of the training and testing sample sets used were respectively, 600 and 400. Surrogates based on genetic programming were trained and tested using this data set. This surrogate was then coupled with NSGA-II to evolve optimal pumping strategies for the coastal aquifer.

5.5 Results and Discussion

5.5.1 Ensemble modelling approach

The inputs and outputs used in the surrogate model development are respectively the pumping rates and the resulting salinity levels. The uncertain parameters are not used

as inputs into the GP surrogate models. Instead, input-output patterns obtained for different combinations of these parameters and these data sets are randomly (random sampling with replacement from the original data pool) used to train the surrogate models. Thus the parameter values are accounted for only implicitly. If the parameters are explicitly used as inputs for the surrogate model development, the methodology may not be scalable for larger applications. As the parameter values are not used in the surrogate model development, the methodology can be extended for heterogeneous values of parameters, although a larger ensemble size may be required for this purpose. However, linking the ensemble surrogate to the optimisation model does not increase the computational burden hugely. Therefore, it is still not impossible to address heterogeneous parameter case with this approach.

This method has definite advantages. It may be illustrated using the following example;

Consider an example with one variable and one parameter with the variable-parameter space represented by the X–Y axes (Figure 5.6). Three different ways of choosing the data domain to train four different surrogate models is shown in figure 5.6(a-c). Choosing data regions *a*, *b*, *c* and *d* to train separate surrogate models will be an inefficient approach because the surrogate will be blind to several combinations, for example, variables in the range of *a* and parameters in the range *c*. Option (b) will be an ideal way to choose the domains for training the surrogates, i.e. train a surrogate model for each parameter value and all variable values in the domain. However, this approach is computationally very expensive and will be difficult to scale for larger applications. Also, when used within an optimisation algorithm, there is a chance that one single surrogate model will determine the binding constraint leading to a chance-constrained like situation. In the third approach, the domain of the data used for training each surrogate model is chosen at random from the variable-parameter domain. For example, consider sub-domains *pvuq*, *pstw*, *xvusr* and *qrxvu* to train four different surrogates. Due to this random choice of the sub-domains, the entire parameter-variable space will be adequately represented when sufficient number of surrogate models are developed. These models will have different predictive capabilities in different regions of the variable-parameter space. Demarcation of sub-domains using straight lines is only for illustration purposes. In reality, this demarcation may be region/regions of data clouds anywhere in the multi-dimensional space.

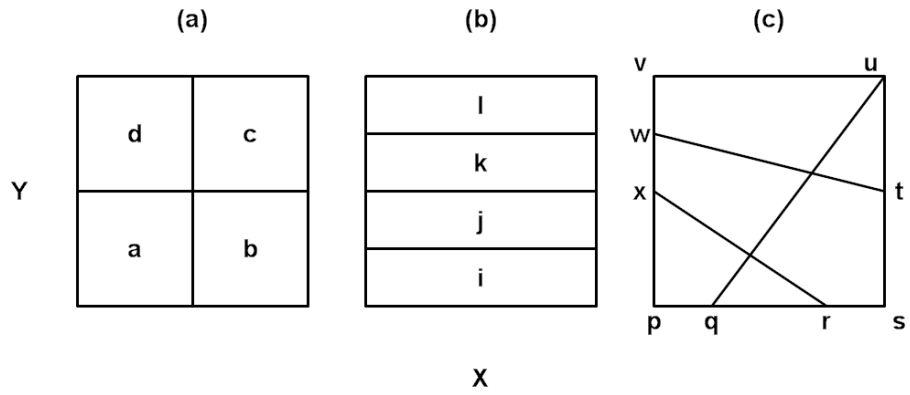

Figure 5.6 Illustration of the decision-parameter sample sets in 2D space

5.5.2 Ensemble surrogate model statistics

Surrogate models were developed to predict the salinity concentration resulting from pumping at specified monitoring locations. Forty such surrogate models were developed for the prediction of salinities at each location C1, C2 and C3. Surrogate models were trained and tested using pumping-concentration data sets which were obtained from FEMWATER models using different realisations of the uncertain parameters. In addition, the prediction is also influenced by the uncertain structure and parameters of the surrogate models. Thus the predictive uncertainty reflects the uncertainties in the groundwater parameters used in the numerical simulation model and the uncertainties in the surrogate model structure and parameters. The salinity concentration values at three nodes, viz, C1, C2 and C3 were monitored in this study. The prediction uncertainty and other analyses were performed for these three locations. All the analyses showed similar results for locations C2 and C3. Hence, for brevity, results corresponding to C1 and C2 are reported in the graphs and tables.

Root mean square error was calculated as an index to evaluate the predictive capability of the surrogate models in the ensemble. Coefficient of variation (CoV) of the RMSEs of the surrogate models in the ensemble was calculated as a measure of the uncertainty of the ensemble. An initial ensemble size of 8 models was considered. The CoV of RMSEs of the predictions by these models was computed. Then surrogate models were added one by one into the ensemble and the CoVs were computed. The CoVs of the RMSEs in the prediction of C1 and C2 for increasing ensemble sizes are plotted in figures 5.7 and 5.8 respectively.

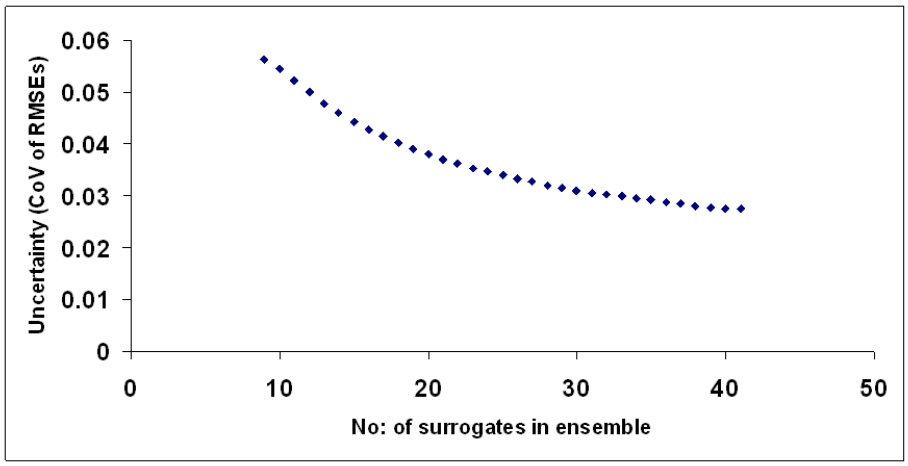

Figure 5.7 Ensemble surrogate model uncertainty for salinity C1

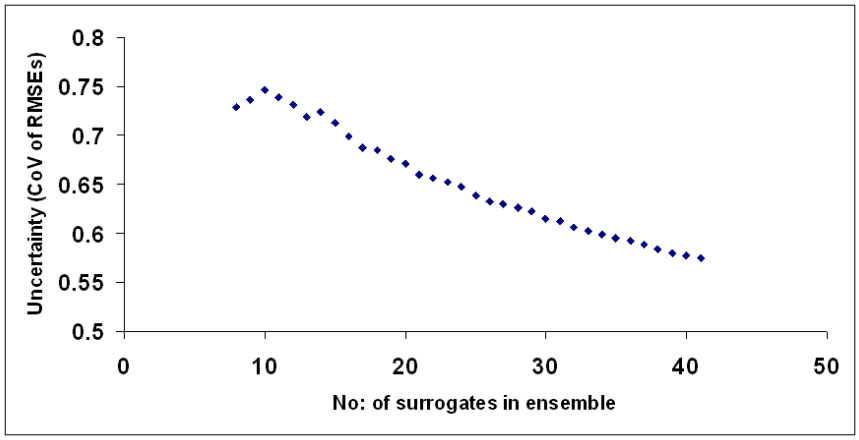

Figure 5.8 Ensemble surrogate model uncertainty for salinity C2

It is observed from figure 5.7 that, for predicting the salinity C1, there is a steady decrease in the value of the CoV of the RMSE with the increase in the number of surrogate models in the ensemble. The CoV is smallest when 37–40 surrogate models are present in the ensemble and it is observed that the corresponding slope of the CoV curve becomes close to zero indicating no further decrease. Therefore, the optimum number of models in the ensemble for predicting C1 was fixed as 40. Similar results were obtained for C2 and C3.

5.5.3 Multi-objective, multiple realisation optimisation

The population size and number of generations used in the NSGA-II algorithm were respectively, 250 and 750. This means that at each stage of optimisation, 250 candidate solutions are evaluated in parallel for their objective function and constraint values. After a number of numerical experiments it was found that a population size of at least 250 is required to obtain the full Pareto-optimal front. When the population

size was smaller, parts of the actual front were eliminated from the final solution. The crossover and mutation probabilities were respectively 0.85 and 0.02. The candidate solutions are crossed over and mutated at these probabilities to generate a new population of solutions. The NSGA-II algorithm uses a tournament selection and simulated binary crossover (Deb 2001). This process is repeated 750 times to obtain the Pareto-optimal set of solutions.

In this study, the ensemble surrogate models are called by the optimisation program for evaluation of the concentration value at the monitoring locations resulting from each scenario of pumping. This means, each surrogate model in the ensemble is individually called by the optimisation routine to predict the concentration level at the monitoring node. Thus, 40 values of concentration are predicted each for C1, C2 and C3. In the multiple realisation optimisation approach. the reliability of the solution is defined as the fraction of the surrogate models which satisfies the imposed constraints. For a reliability of 0.8, 32 out of these 40 surrogate model predictions satisfy the imposed constraints.

The coastal aquifer management problem has 33 variables corresponding to pumping from 11 locations for three time periods. The multiple realisation approach calls 120 different surrogate models, 40 each corresponding to the salinity concentrations C1, C2 and C3, during the evaluation of each candidate solution.

5.5.4 Optimal solutions with different reliabilities

An optimal solution to the pumping optimisation problem considering two conflicting objectives is a Pareto-optimal front of solutions which defines a trade-off between the two objectives. The Pareto-optimal front obtained, when the constraints imposed by all the 40 surrogate models in the ensemble are satisfied in the optimisation, is considered as the solution corresponding to a reliability level of 0.99. Similarly, optimal solutions with reliabilities 0.8 and 0.6 satisfy 32 and 24 out of the total 40 models, respectively. The Pareto-optimal set of solutions corresponding to these different reliability levels is shown in 5.9. This figure also shows the Pareto-optimal front corresponding to the single surrogate model based optimisation. This front appears to deliver better optimal solutions. However, when the corresponding pumping values are input in the numerical simulation model, it was observed that the all the solutions lying in the front violate some of the constraints. The predicted

concentration values and corresponding constraint violations for five solutions from different regions of the front are shown in table 5.2.

Table 5.2 Optimal solutions and constraint violations using a single surrogate model

Solution	Total pumping ($\times 10^3$ m^3/d)		Constraint		($\times 10^{-3}$ kg/m^3)	
	Beneficial	Barrier	C1 (<500)	Violation	C2 (<600)	Violation
1.0	251.6	15.1	544.3	44.3	765.5	165.5
2.0	204.9	9.4	537.6	37.6	688.7	88.7
3.0	232.6	10.6	545.6	45.6	758.9	158.9
4.0	284.8	27.1	534.7	34.7	773.3	173.3
5.0	310.3	55.5	510.0	10.0	710.3	110.3

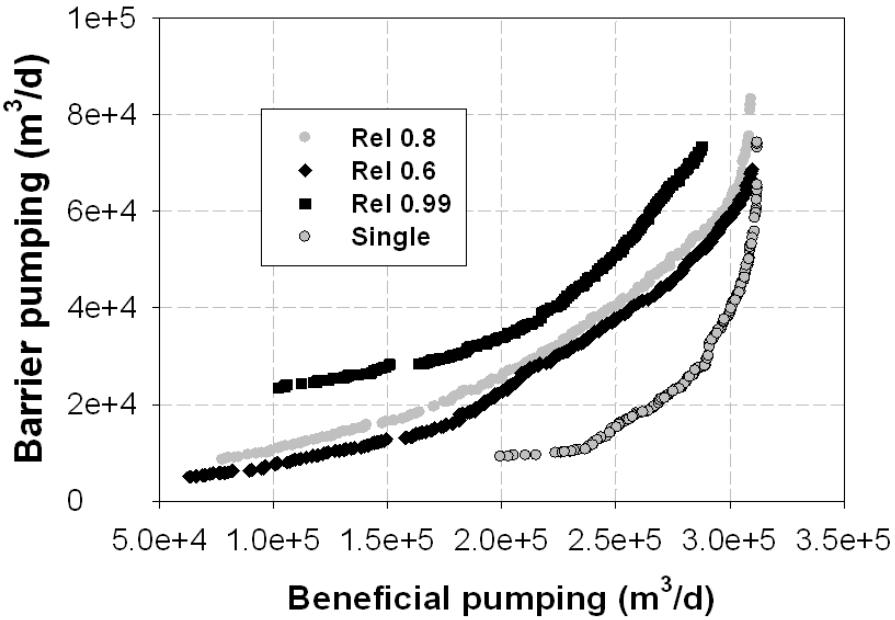

Figure 5.9 Pareto-optimal fronts for different reliabilities and single surrogate modelling approach

It was observed that this front actually represents solutions in the infeasible domain. This is primarily because of the wrong predictions by the single surrogate model. However, it was found that the RMSE values for the single surrogate models for C1, C2 and C3 were similar to that of the surrogate models in the ensemble. Thus the average prediction accuracy does not indicate accurate levels of prediction for the optimal solution. This is because in an attempt to search for the optimal values of the objective functions, the optimisation algorithm chooses those candidate solutions

which give the highest positive error (i.e. surrogate model prediction – numerical model observed) in the prediction of concentration. The probability distribution of the errors for C1 and C2 obtained from the single surrogate model are plotted in figures 5.10 and 5.11. From table 5.2 and these two figures it is evident that the errors in the concentration prediction for the optimal solutions belong to the positive tail end of the distribution and beyond.

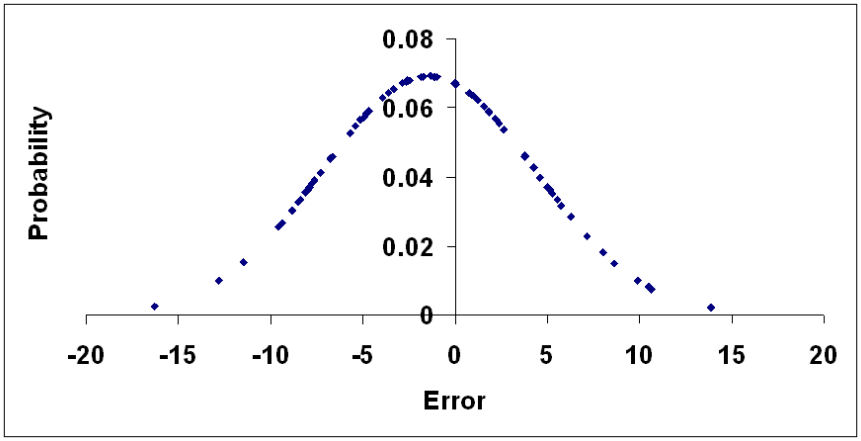

Figure 5.10 Prediction error distribution for C1

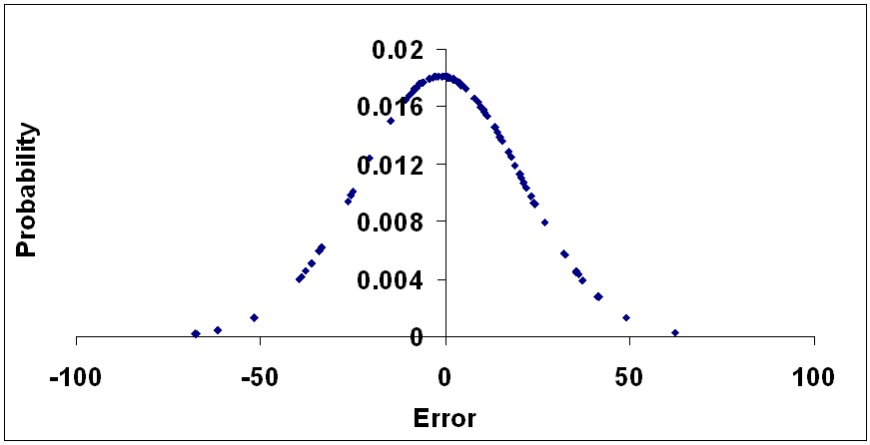

Figure 5.11 Prediction error distribution for C2

From this we deduce that an original data set with 250 patterns is insufficient to train and validate a single surrogate model which satisfactorily predicts the concentrations in all regions of the parameter-decision space. It may not be impossible to develop a single surrogate model capable of doing this, but it may require an exponentially large number of training and testing patterns of pumping-salinity data. However, developing a surrogate model with such a huge training data set may defy the objective of obtaining computational efficiency using surrogate models. If uncertainty in a larger number of numerical model parameters is

considered, a single surrogate may be infeasible as it may require a large number of patterns to train the surrogate model involving a huge computational burden. This is because multiple uncertain parameters in the numerical model will be mapped in to a single set of surrogate model structure and parameters. A large number of surrogate model parameters may be required for this and may result in over-fitting of the model.

Multiple realisation optimisation with ensemble surrogate models is found to be an efficient methodology for evolving reliable optimal solutions in the Pareto-optimal front. The Pareto-optimal fronts for the reliability levels 0.99, 0.8 and 0.6 are shown in figure 5.9. The objective function values corresponding to reliability level of 0.99 appear to be the least optimal value amongst all the fronts. However, it will be closer to the actual Pareto-optimal front because for the lesser reliability levels, at least some of the solutions in the front move into the infeasible region as they violate the constraints. Twenty-five solutions chosen from different regions of each front were cross-checked for their constraint violation, using the numerical simulation model. The average values of constraint violation corresponding to three different reliability levels and three different combinations of parameter values are shown in table 5.3. Five points from each front, with corresponding objective function values and constraint values for three different realisations of the uncertain parameters are shown in tables 5.4, 5.5 and 5.5. These realisations are obtained from different regions of the respective distributions for those uncertain parameters. The solutions with a reliability level 0.99 exhibit virtually no constraint violation, where a violation within 1% excess of the prescribed limit of concentration is considered acceptable. Also, it is observed that as the reliability level decreases the constraint violation increases.

Table 5.3 Average values of constraint violation by solutions of different reliabilities

Hydraulic conductivity, Recharge	32.67 m/d, 0.176 m/yr		24.44 m/d, 0.061 m/yr		45.59 m/d, 0.258 m/yr	
	Constraint violation (% of the concentration limit C1 = 500 and C2 = 600)					
Reliability	C1 (<500)	C2 (<600)	C1 (<500)	C2 (<600)	C1 (<500)	C2 (<600)
0.99	-0.13	0.00	-0.35	0.00	-0.05	0.00
0.8	-1.35	-1.44	-1.63	-1.80	-0.49	0.00
0.6	-2.20	-3.56	-2.96	-3.90	-2.40	-3.45

Negative value indicates a constraint violation

Table 5.4 Concentrations corresponding to optimal solutions with reliability 0.99 obtained from FEMWATER with different values of uncertain parameters

Hydraulic conductivity, Recharge	Objectives		32.67 m/d, 0.176 m/yr		24.44 m/d, 0.061 m/yr		45.59 m/d, 0.258 m/yr	
	Total pumping ($\times 10^3 m^3/d$)		Constraint ($\times 10^{-3} kg/m^3$)		Constraint ($\times 10^{-3} kg/m^3$)		Constraint ($\times 10^{-3} kg/m^3$)	
Solution No:	Production	Barrier	C1 (<500)	C2 (<600)	C1 (<500)	C2 (<600)	C1 (<500)	C2 (<600)
1	285.7	69.3	485.2	564.1	502.95	584.92	481.10	550.18
2	228.9	43.3	502.9	577.9	501.53	590.42	499.21	555.25
3	175.5	30.1	503.0	563.7	501.22	582.87	499.24	551.43
4	105.9	23.7	485.1	565.6	488.09	575.92	478.96	550.15
5	134.3	25.4	498.4	555.6	501.11	562.58	492.11	545.86

Table 5.5 Concentrations corresponding to optimal solutions with reliability 0.8 obtained from FEMWATER with different values of uncertain parameters

Hydraulic conductivity, Recharge	Objectives		32.67 m/d, 0.176 m/yr		24.44 m/d, 0.061 m/yr		45.59 m/d, 0.258 m/yr	
	Total pumping ($\times 10^3 m^3/d$)		Constraint ($\times 10^{-3} kg/m^3$)		Constraint ($\times 10^{-3} kg/m^3$)		Constraint ($\times 10^{-3} kg/m^3$)	
Solution No:	Production	Barrier	C1 (<500)	C2 (<600)	C1 (<500)	C2 (<600)	C1 (<500)	C2 (<600)
1	120.3	11.3	510.14	585.33	515.28	602.30	514.58	579.13
2	303.2	65.9	505.39	584.55	508.20	592.22	498.78	564.03
3	207.6	27.9	510.54	615.10	511.46	619.79	503.82	595.15
4	261.6	44.9	511.78	624.89	514.41	625.20	508.47	597.33
5	161.3	18.4	505.98	601.91	507.30	608.93	495.54	583.50

Table 5.6 Concentrations corresponding to optimal solutions with reliability 0.6 obtained from FEMWATER with different values of uncertain parameters

Hydraulic conductivity, Recharge	Objectives		32.67 m/d, 0.176 m/yr		24.44 m/d, 0.061 m/yr		45.59 m/d, 0.258 m/yr	
	Total pumping ($\times 10^3 m^3/d$)		Constraint ($\times 10^{-3} kg/m^3$)		Constraint ($\times 10^{-3} kg/m^3$)		Constraint ($\times 10^{-3} kg/m^3$)	
Solution	Production	Barrier	C1 (<500)	C2 (<600)	C1 (<500)	C2 (<600)	C1 (<500)	C2 (<600)
1	201.6	24.1	497.94	592.02	508.33	599.82	492.69	578.29
2	248.9	37.4	518.61	649.46	519.45	648.17	510.73	637.32
3	277.2	48.2	513.31	633.40	515.67	641.56	505.48	617.12
4	309.4	68.1	500.56	590.94	511.29	595.80	493.28	578.39
5	172.5	15.5	517.93	621.31	515.63	627.34	542.81	648.93

In order to investigate the sensitivity of the optimal solutions to the optimal number of surrogate models in the ensemble, numerical experiments were conducted using five different optimisation models with 38, 39, 40, 41, 42 and 46 models in the

ensemble. All these models evolved the same Pareto-optimal front. However, arbitrarily increasing the ensemble size to a great extent may produce a front which is sub-optimal. This is because, as the ensemble size increases complexity of the optimisation problem increases as well. This may prevent the convergence to a global optimal solution.

The methodology has specific advantages over the single surrogate modelling approach. The primary advantage is that this methodology can, to a certain extent, address the uncertainties in groundwater parameter values used in the numerical model. It also has the advantage of reducing the number of patterns required for developing the surrogate models.

Due to the random selection of patterns with repetition into the bootstrap samples, different bootstrap sample sets have different weighting for different regions of the decision space. Depending on the Latin Hypercube Samples of uncertain parameters used in the numerical model to simulate the concentration corresponding to these patterns, these patterns have different weighting for different regions in the parameter space too. When a number of such models are used, they adequately represent the total decision-parameter space. This is ensured using Latin Hypercube Samples of the pumping patterns, hydraulic conductivity and recharge values. In the multiple realisation approach, optimal pumping strategy will not be dictated by one single “worst” realisation (Feyen & Gorelick 2004). The most active constraint in different iterations of the optimisation will be different.

It is possible to train a surrogate model based on a single set of most constraining uncertain parameters. For example, if one considers the least possible annual recharge value and the least hydraulic conductivity realisation, the worst constraining surrogate model can be developed. If such a surrogate model is 100% accurate, it would result in the most conservative pumping solutions. However, if the predictive uncertainty of the single surrogate model is considered, it would result in infeasible solutions. This is because, as discussed earlier, the error in the concentration values for optimal solutions corresponds to the positive tail end of the concentration error distribution. For example, if we consider an error distribution as shown in figure 5.10 or 5.11, all the concentration values predicted when evaluating the objective function and constraints during optimisation will have a random value from this distribution. In an attempt to maximise the production well pumping and minimise the barrier well pumping, the optimisation algorithm tends to choose those

surrogate model predictions which actually under predicts the saltwater intrusion. Thus, near the optimal solutions the algorithm always chooses surrogate model predictions which have an error corresponding to the positive tail end (under prediction) region of the error distribution. These high error values will push the optimal solutions into the infeasible regions. In the multiple realisation approach, different constraints are active at different stages. This helps to evolve optimal solutions which are actually in the feasible domain.

This study utilises pumping-salinity patterns generated using different sets off uncertain parameters to train each surrogate model. It is also possible to train each surrogate model using a single set of uncertain parameters. This approach was not adopted in this work because we believe that if such an ensemble is generated with each surrogate model corresponding to a single set of uncertain parameters; it may result in a single "worst" constraint in spite of the presence of multiple constraints in the optimisation model. This would result in conservative, sub-optimal or even infeasible solutions.

5.5.5 Robustness of optimal solutions

The validation of the optimal solutions using FEMWATER for uncertain parameters chosen from different regions of the parameter space, as shown in tables 5.4, 5.5 and 5.6, indicate that the solutions obtained using the multiple realisation optimisation are also robust optimal solutions. The concentration values obtained for the optimal solutions for different values of the parameters are very close to each other. It could be concluded that when multiple constraints are incorporated in optimisation using different surrogate models, the algorithm searches for solutions which simultaneously satisfy the constraints imposed by all the surrogate models. This results in robust optimal solutions.

5.6 Summary and Conclusion

An ensemble surrogate modelling approach together with multiple realisation optimisation is proposed for multi-objective pumping management of a coastal aquifer. The proposed methodology is demonstrated with application to a coastal aquifer system in the Burdekin delta area in Australia. The methodology considers uncertainty in two groundwater parameters viz, hydraulic conductivity and annual aquifer recharge. Latin Hypercube Samples of the uncertain parameters and pumping scenarios were input in the three-dimensional finite element based numerical

simulation model FEMWATER to predict the salinity concentrations at specified monitoring locations. The resulting pumping-salinity patterns constitute an original set of data for developing the surrogate models. Bootstrap samples of these data sets were generated by sampling with replacement from the original data set. The bootstrap sample sets were used to train and test different surrogate models for predicting the concentrations at the monitoring locations. Coefficient of variation of the RMSEs was used as a measure of predictive uncertainty of the ensemble surrogate models. The surrogate models in the ensemble were independently linked to the multi-objective optimisation algorithm to solve the pumping management problem considering two conflicting objectives. Maximisation of pumping from beneficial wells and minimisation of the total pumping from barrier wells were the two objectives. Reliability of the optimal solutions was measured as the fraction of the surrogate model predictions which satisfy the upper limit on the concentration defined in the optimisation model. On validation of the solutions with the actual numerical model it was found that the solutions with reliability 0.99 satisfy all the constraints on the concentration limit. With decreasing reliability levels there is an increase in the number of violations.

Ensemble surrogate models coupled with multiple realisation optimisation has definite advantages over the single surrogate modelling approach. It helps to better account for the uncertain groundwater parameters. Most often the errors in the surrogate model predictions for optimal solutions correspond to high values in the tail end regions of the error distribution, for objective functions like maximisation of pumping. Hence, using single surrogate models to substitute numerical simulation models may result in infeasible or sub-optimal solutions. Ensemble surrogate models with multiple realisation optimisation helps to overcome this problem to a certain extent. The ensemble approach also reduces the computational requirement by reducing the number of patterns required to train and test the surrogate models. Thus the proposed ensemble surrogate modelling approach is a computationally efficient methodology that can be used to solve groundwater management problems considering uncertainty in groundwater parameters. The results also indicate that the solutions obtained are robust in nature.

As a first attempt to address groundwater parameter uncertainty within the optimisation model using ensemble surrogate models, this work considered only a homogeneous value for the uncertain hydraulic conductivity and aquifer recharge.

The methodology may be tested for its robustness considering a heterogeneous hydraulic conductivity field. Similarly, uncertainty in the transport parameters may also be addressed using this methodology. However, it is foreseen that it may require a larger number of surrogate models in the ensemble when larger uncertainties in multiple parameters are considered. Also, it may be difficult to increase the ensemble size beyond a certain limit as it can cause computational instability and non-convergence. Training and testing very large number of surrogates may also defy the computational advantage of the surrogate modelling technique. These are some limitations of the proposed methodology that need to be addressed in future studies.

6. DESIGN OF OPTIMAL COMPLIANCE MONITORING NETWORK AND INCORPORATING FEEDBACK INFORMATION FOR COASTAL AQUIFER MANAGEMENT

A similar version of this chapter has been submitted to *Journal of Water Resources Planning and Management.*

6.0 Overview

Methodologies for obtaining management strategies for the optimal and sustainable use of groundwater resources are developed in the previous chapters. These methodologies are based on prescriptive models which utilise mathematical tools for simulation and optimisation together with field data. Methodologies for optimal management of aquifers prescribe the strategies that need to be implemented in the field. Often, the actually implemented pumping strategy deviates from the prescribed ones. These deviations may result from the non-compliance of the users with a prescribed plan. Or, even if the prescribed strategies are implemented exactly, the uncertainties associated with the prediction of impacts results in actual field scale deviations from the predicted or intended impacts on salinity control. Due to the uncertainties inherent in the groundwater systems, it is essential to verify the compliance of the implemented strategies to those prescribed and also verify the impacts of an implemented strategy by using proper monitoring techniques during and post implementation stages of the groundwater management project. In this work, a methodology is developed to design an optimal network to monitor the compliance in terms of the salinity concentration levels which results from the implementation of a planned sustainable pumping strategy for coastal aquifer management. Uncertainty in the values of groundwater parameters and the uncertainty due to the deviation of the pumping strategies from the prescribed optimum values are characterised by considering different realisations of these values in the 3D density-dependent flow and transport simulation model. A new objective of monitoring is considered in this study. The objective function comprises of maximising the coefficient of variation of the salinity concentration at the monitored locations, and minimising the correlation coefficient between the concentrations at the monitored locations. Using this objective, monitoring locations are chosen in regions where the uncertainty in the

concentration values is maximum. Those locations are also chosen where the correlation between the concentrations of the monitored locations is minimum so that the redundancy in monitoring data is the least. The concentration data collected at the optimal compliance monitoring locations can be used as feedback information to improve or modify the initially developed coastal aquifer management strategies.

6.1 Monitoring network design

Two different monitoring network design objectives are considered. In both the monitoring network designs a number of potential monitoring locations are considered. Ideally for any groundwater system, monitoring information from as many locations as possible may be collected to better characterise the system, thereby reducing the uncertainty. This helps in developing accurate solutions for groundwater management. However, in practice, this is seldom possible because of the budgetary constraints on monitoring. Hence, all the designs proposed here are constrained by an upper limit on the number of possible monitoring wells that can be installed.

6.1.1 Monitoring network design I

The objective considered in monitoring design I is to install monitoring wells in those potential locations where the uncertainty, quantified in terms of the coefficient of variation of estimated concentrations, is maximum. The mathematical formulation of the monitoring design I is as follows:

$$\textit{Maximize} \quad \sum_{t=1}^{T}\sum_{i=1}^{N}\frac{\sigma_{C_i^t}}{\overline{C_i^t}}\times X_i^t \tag{6.1}$$

$$\sigma_{C_i^t}=\frac{\left(C_i^t-\overline{C_i^t}\right)}{N\overline{C_i^t}} \tag{6.2}$$

subject to:

$$\sum_{t=1}^{T}\sum_{i=1}^{N}X_i^t \le W \tag{6.3}$$

$$X_i \in \{0,1\} \tag{6.4}$$

where, C_i^t is the concentration realisations at the i[th] location at the t[th] time period, $\overline{C_i^t}$ is the mean value of the concentration at the i[th] location at the t[th] time period, $\sigma_{C_i^t}$ is the variance of the concentration at the i[th] location at the t[th] time period, *N* is the total number of potential monitoring locations, *T* is the total number of time steps, and W is the maximum number of monitoring wells permitted within the budgetary constraints. X_i^t is either 1 or 0 indicating installation or non-installation of a monitoring well, respectively.

6.1.2 Monitoring network design II

Monitoring network design II is mathematically formulated as follows:

$$Maximize \sum_{i,j \in N} \sum_{t \in T} \left\{ \frac{\sigma_{C_i^t}}{\overline{C_i^t}} - \frac{COV(C_i^t, C_j^t)}{\sigma_{C_i^t} \sigma_{C_j^t}} X_j^t \right\} X_i^t \tag{6.5}$$

The same constraint as in (6.3) and (6.4) applies to monitoring design II. In this objective function, the first term facilitates the placement of monitoring wells at locations of maximum uncertainty in the estimated concentration value. The second term in this objective function evaluates the correlation coefficient between the values of concentration for all monitoring locations. In order to obtain a maximum value of the objective function, the second term needs to be minimised. Those locations which have the least correlation value amongst all the locations of maximum uncertainty need to be chosen for this. C_i^t and C_j^t are concentrations at the potential monitoring locations *i* and *j* at time *t*. X_i^t and X_j^t are binary variables, i.e. either 1 or 0 indicating installation or non-installation of a monitoring well. The optimal monitoring network design problems are combinatorial optimisation type and were solved using LINGO 13 (LINDO 1999).

6.2 Monitoring feedback-based updating of optimal management strategies for coastal aquifers

To evaluate the utility of the compliance monitoring in coastal aquifer management, the monitoring network design methodology is applied to a single objective

management problem. The single objective optimisation maximises the total pumping from a coastal aquifer well field subject to controlling the resulting salinity levels at pre-specified limits. Limit on the total pumping from barrier wells in the saltwater wedge region is incorporated as an additional constraint. These barrier wells are utilised to control saltwater intrusion near the sea-face boundary to specified feasible limits by hydraulic control of the flow gradient. The mathematical formulation is given as follows:

$$Maximize \sum_{i=1}^{N}\sum_{t=1}^{T} Q_i^t \tag{6.6}$$

$$\text{subject to } C_j^t \le C^{\max} \tag{6.7}$$

$$\sum_{k=1}^{K}\sum_{t=1}^{T} q_t^k \le q_{\max} \tag{6.8}$$

$$Q_{\min} \le Q_i^t \le Q_{\max} \tag{6.9}$$

$$q_{\min} \le q_k^t \le q_{\max} \tag{6.10}$$

A coupled simulation optimisation approach using genetic programming based ensemble surrogates, as described in chapter 5, is used to solve the coastal aquifer management problem. However, to demonstrate the sequential modification of the field implementation of the strategies, the multi-objective optimisation problem is converted into a single-objective problem by imposing a constraint on the total barrier well pumping. A single objective genetic algorithm (Deb 2001) is used to solve the optimisation problem. The focus of this chapter is the design of the optimal compliance monitoring network for evaluating the actual implementation of the strategy and to sequentially update the pumping strategies after each time-step of operation. These sequential modifications are based on the compliance information gathered from the designed and implemented monitoring network. This approach addresses any non-compliance that may have occurred in the previous time step of operation. Compliance is defined in terms of meeting the set salinity constraints. The monitoring network helps to better estimate the spatial and temporal concentrations to ensure that the set goals are satisfied.

A scientifically designed compliance monitoring network can be used to collect data which gives information on the compliance or non-compliance of a designed optimal pumping strategy in the field. Most of the optimal strategies for

groundwater management are developed for long time frames of the order of many years and are implemented in stages. With a properly designed compliance monitoring network it is possible to get feedback information at each time-step t within the entire management time horizon T. This feedback information collected at each time-step can be used to update the management strategies for the subsequent time-steps of operation if any deviation from the predicted salinity levels is observed at the end of any stage of the implementation.

This compliance monitoring network can be used to monitor the saltwater intrusion levels at any stage during and post implementation stages of the optimal operation strategies. This feedback information can be used to modify or improve the pumping strategies for the subsequent time-steps of operation considered within the management framework. This methodology is illustrated using numerical simulation experiments.

Optimal pumping management for a coastal aquifer with a management time frame of three years is considered. The monitoring network is designed for a specified pumping strategy which corresponds to a particular solution obtained from the simulation-optimisation model. However, the designed network is robust to the uncertainties arising due to the groundwater parameter uncertainty and field level deviation in the implementation of prescribed pumping strategies. This is ensured by considering Monte Carlo simulations based on different realisations of these values in the design of an optimal compliance network model. A number of realisations of hydraulic conductivity and deviations in the implementation of the optimal strategy are considered in the optimal design of the monitoring network.

The deviation in the field implementation of optimal strategies for the first year of operation is simulated numerically using the coupled flow and transport model FEMWATER by adding random errors to the optimal pumping values. The random errors correspond to the inadvertent random deviations from the prescribed strategy during the field implementation. These random deviations affect the salinity concentrations at the optimal monitoring locations, and may result in non-compliance with the expected concentration profile at the end of first year. For non-compliance scenarios, the concentration levels that would result at the compliance monitoring locations at the end of the entire management horizon are obtained using FEMWATER.

By using the proposed methodology which uses feedback information from the optimal monitoring locations to evaluate the compliance, the prescribed pumping strategy for the subsequent time steps of management is modified or improved to address the deviation from the initially prescribed optimal strategy for the first time period of operation. This is obtained using the simulation-optimisation approach which is updated with the information available at the end of first year of operation. This approach is repeated for all time-steps of operation considered within the management time period.

The monitoring feedback based optimal management of coastal aquifers is compared against the management framework which implements only the initial prescribed strategies irrespective any deviations observed in the previous stages. The comparison is in terms of the compliance observed at the end of the management time horizon. Non-compliance is measured as the deviation from the expected concentration levels as per the initial optimal strategy. The objective function values, i.e. the total beneficial pumping, obtained from these two approaches are also compared.

6.3 Performance evaluation

In this study, the developed optimal compliance monitoring network design formulation and the feedback-based optimal management strategy is tested for a realistic coastal aquifer system in the Burdekin region of Queensland, Australia which is described in chapter 5.

6.3.1 Study area

The considered study area is 60.2 km^2 in areal extent and is located near the coast. In this region, water from an unconfined aquifer with an average thickness of 60 metres is extracted mainly for irrigating sugarcane. This study area is bounded on two sides by the Burdekin River and by the coastline on the east. Figure 6.1 illustrates the study area with the location of the production and barrier wells. Two types of wells are considered in this study. The first one is a set of 8 production wells, which represent regionally averaged beneficial pumping from the area and the second type is a set of 3 barrier wells which pump out water from the saltwater wedge region for hydraulically controlling saltwater intrusion. The barrier wells are indicated in solid black in figure 6.1. Uncertain values of hydraulic conductivity and annual aquifer recharge are considered in deriving optimal solutions. In the present work, design of an optimal

monitoring network is presented for monitoring the compliance of salinity levels at the end of each time-step of implementation of a pumping strategy.

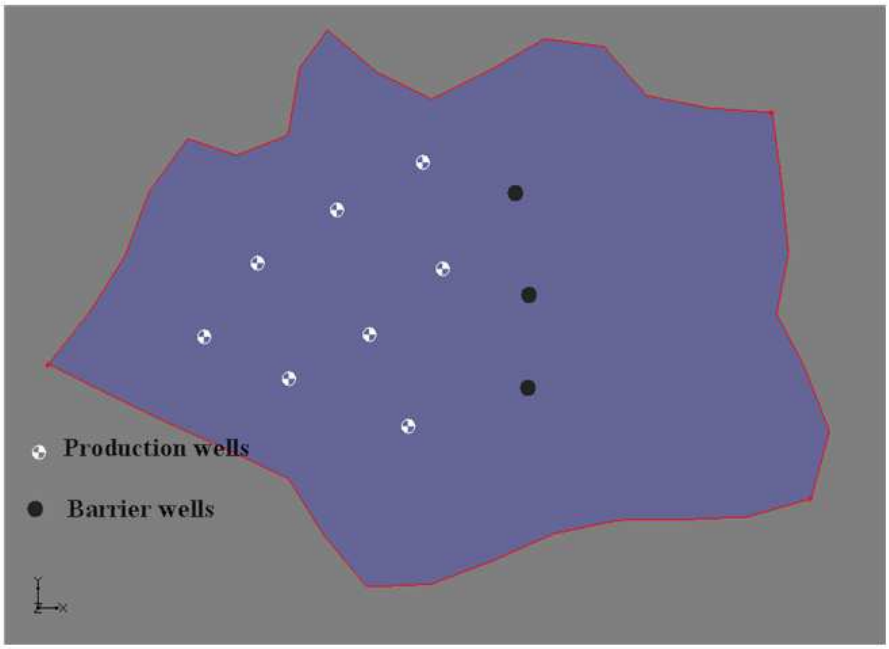

Figure 6.1 Plan view of the study area with location of wells

6.3.2 Uncertainty characterisation

Different possible realisations of the salinity concentration profile resulting from the implementation of a specific strategy of pumping from the coastal aquifer are used in the optimal design of the compliance monitoring network. These different possible concentrations profiles result from the uncertainty in the parameters used in the simulation model. In this work, uncertainty in hydraulic conductivity and annual aquifer recharge are considered. Also, implementation of pumping strategies which deviate from the actual prescribed values can also cause variation in the concentration profile from the expected values. The designed monitoring network should be robust enough to accommodate these deviations.

Uncertainty in the hydraulic conductivity values is accounted by sampling a number of possible values of hydraulic conductivity from its log-normal distribution. Latin Hypercube Sampling was adopted to maintain equi-probable representation. Similarly, Latin Hypercube Samples of normally distributed aquifer recharge values are sampled. These values of hydraulic conductivity and aquifer recharge values were randomly paired with each other to characterise the uncertain parameter space used in the modelling. The deviation from the prescribed values of pumping is accounted for by perturbing the values of pumping in its local neighbourhood to obtain different

realisations of pumping. These values of pumping and different realisations of the uncertain parameters are used in a Monte Carlo framework in the FEMWATER flow and transport modelling to develop different realisations of the salinity concentration profile. A hundred different realisations of the concentration profiles were developed by using random combinations of uncertain parameters and statistically perturbed pumping values.

6.3.3 Potential and permissible number of monitoring locations

Ideally any location within the study area could be a location for a potential monitoring well. In the finite element model for simulating the groundwater flow and transport processes, any node can be a potential monitoring location. However, the nodes in the freshwater region may not be included as potential monitoring locations. In this study, 96 potential monitoring locations were considered. All finite element nodes falling within a concentration range of 100–17000 mg/L, in at least one realisation, were considered as potential monitoring locations. The potential monitoring locations are shown in figure 6.2. The budgetary limit on monitoring is implicitly accounted by limiting the maximum number of monitoring wells. An initial maximum number of permissible wells was considered as 10. Another design with 20 as the maximum number of permissible wells is also presented.

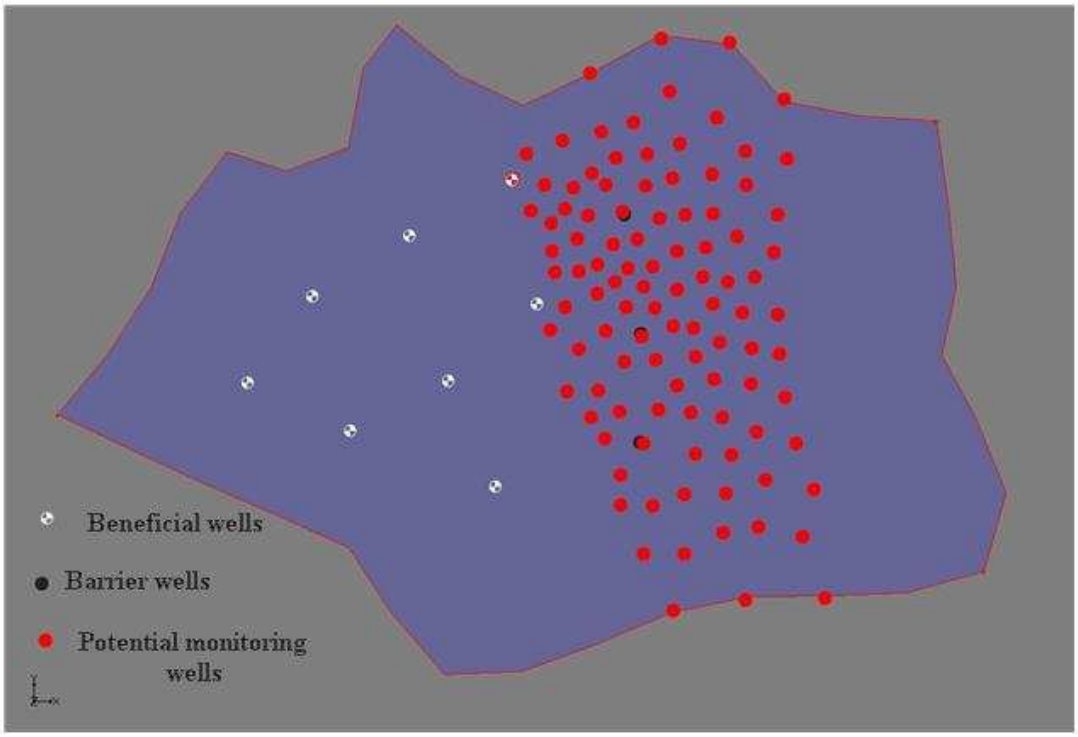

Figure 6.2 Potential monitoring locations

6.4 Results and Discussion

6.4.1 Monitoring network design

Monitoring design I was used to design the pilot compliance monitoring network for the Burdekin study area with two different scenarios based on the maximum number of monitoring wells that can be installed. The optimal monitoring network designs with a maximum of 10 wells based on the monitoring network design I and II are shown in figures 6.3 and 6.4 respectively. The designs for 20 wells using monitoring network design I and II are shown in figures 6.5 and 6.6.

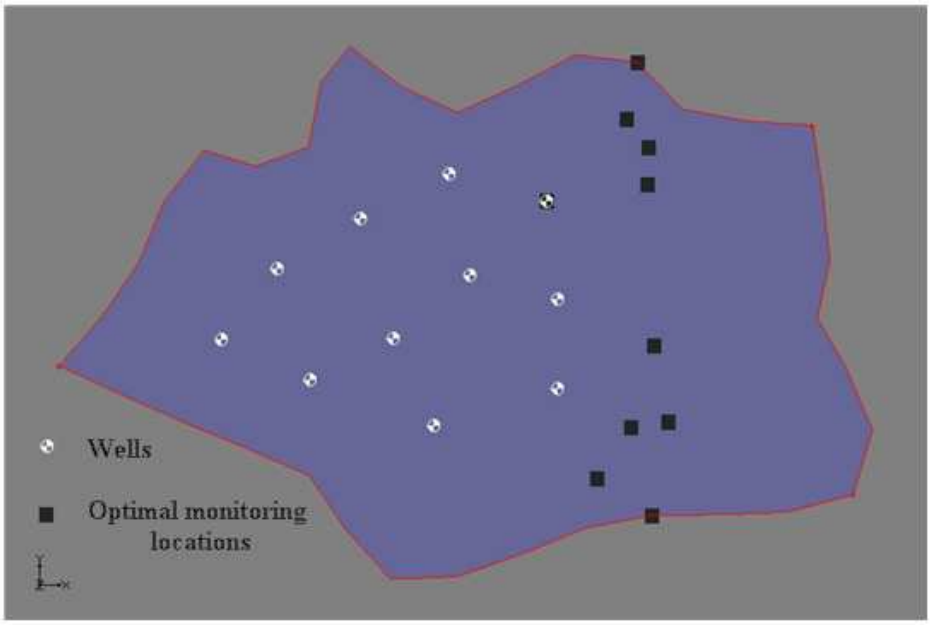

Figure 6.3 Monitoring network design I (10 wells)

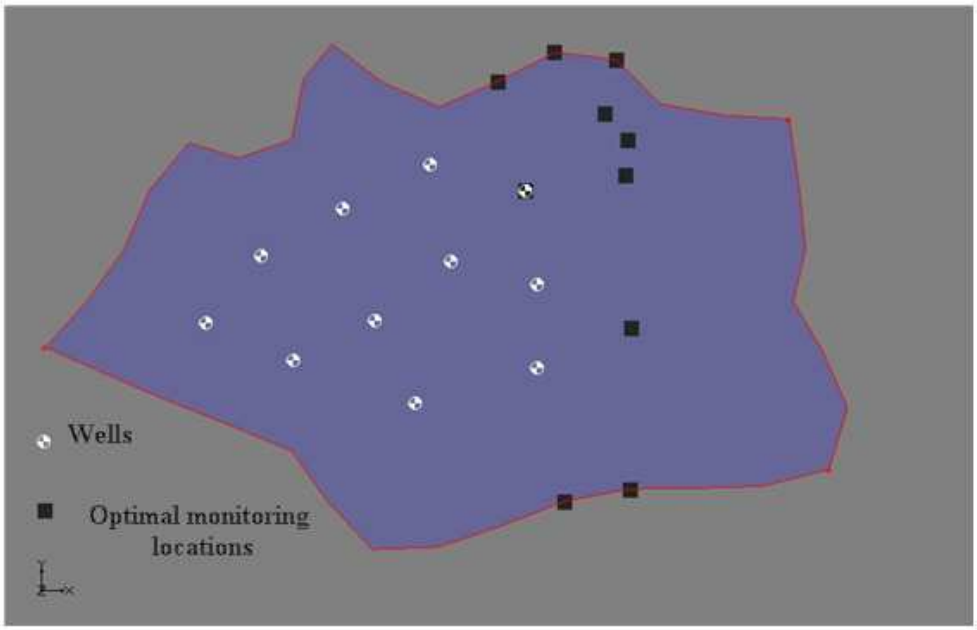

Figure 6.4 Monitoring network design II (10 wells)

Monitoring network design I chooses to install monitoring wells at potential monitoring locations based only on the uncertainty in the values of concentration measured at these locations in terms of the concentration variance. From figures 6.3 and 6.5 it could be observed that, there is a high density of monitoring locations in a specific region where the variance is high. As the uncertainty in this region is high it is necessary to have a larger number of monitoring wells in this region. However, placing a large number of wells in this region would result in redundancy in the information collected and a compromise on the information at some other regions. In figure 6.5 clustering of monitoring wells at a specific region can be observed.

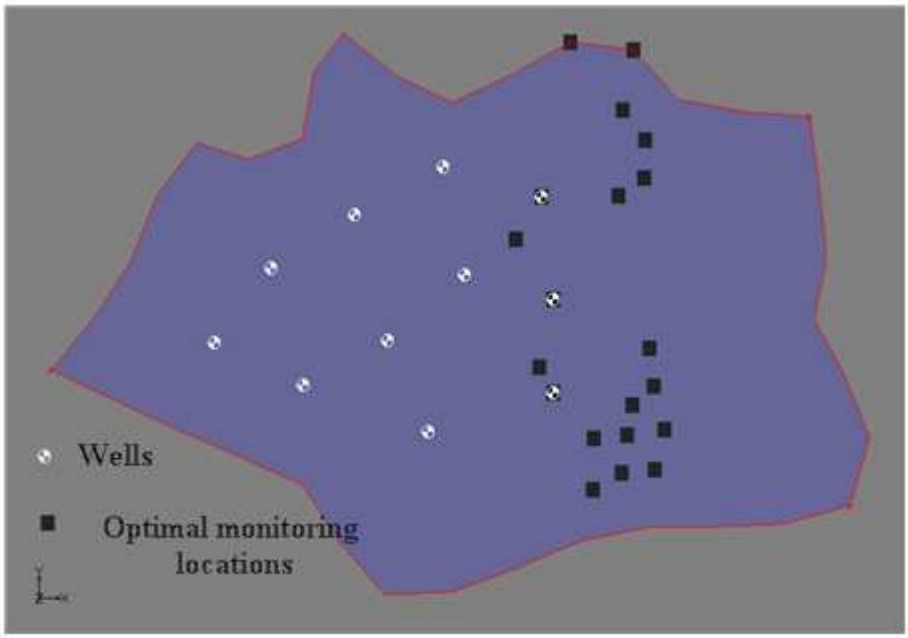

Figure 6.5 Monitoring network design I (20 wells)

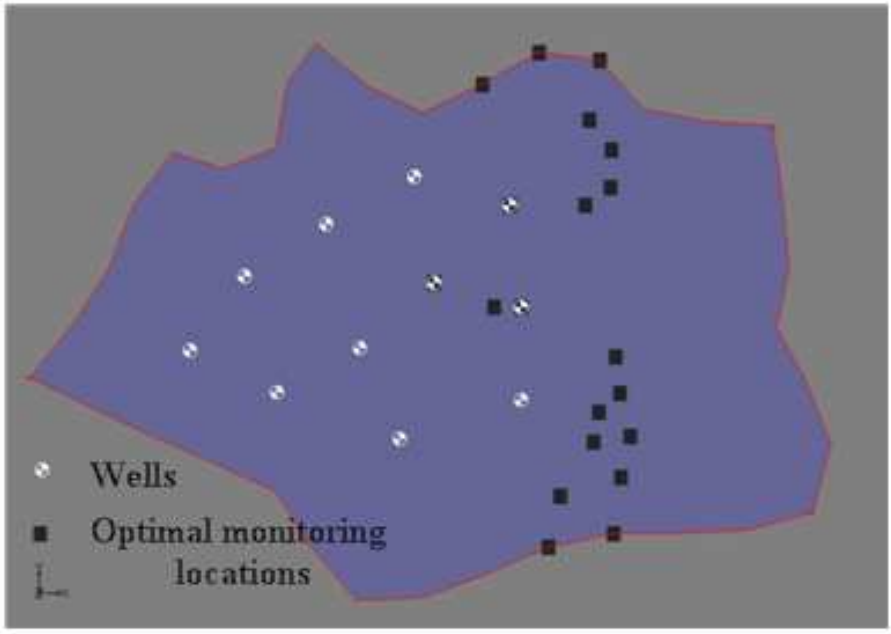

Figure 6.6 Monitoring network design II (20 wells)

Monitoring network design II incorporates an additional criterion in terms of the concentration correlation between monitored locations. This design also chooses locations where the uncertainty is maximum as the first component of the objective function indicates. However, in doing so, the second component of the objective function minimises the correlation between the concentration values at the chosen monitoring locations. Thus, those potential locations with large uncertainty levels and

low correlation between each other are chosen. As the concentration covariances, and hence, the correlation are spatially dependent, this helps in preventing the clustering of the chosen monitoring wells in specific regions. Comparing the figures 6.3 and 6.4 and 6.5 and 6.6, it could be observed that the monitoring network design II is useful in dispersing the monitoring wells from each other to a certain extent. Thus, design II helps in reducing the redundancy in monitoring locations. Also, the spread of the monitoring locations in space was quantified in terms of the sum of the distance of each monitoring location to the co-ordinates of their centroid. For monitoring network design I with 20 wells, the centroidal distance was 38617 metres and for design II the corresponding value was 43220 metres. Evidently, the proposed new formulation for monitoring network design helps in reducing the redundancy in monitoring information resulting from the spatial correlation of the concentrations.

6.4.2 Initial optimal solutions for pumping management

Based on the linked simulation-optimisation model, an optimal pumping strategy was obtained for the operation of the coastal aquifer well field for a three-year management horizon. This is considered as scenario 1. The total pumping from the barrier wells was limited to be less than 35000 m^3/d. For the selected optimal pumping strategy, the total pumping was 219842 m^3/d. The 33 values of optimal pumping rates corresponding to pumping from 11 locations for three time periods are shown in figure 6.7. The variables are named as P*m_n* and B*m_n* where P and B refer to “production” and “barrier” wells and *m* and *n* refer to the year of operation and well number respectively. For example, P1_2 refers to pumping rate from production well 2 in the first year of operation. Once the management model is solved for a three-year time horizon, the pumping strategy prescribed for the first year is implemented. Thereafter, based on the monitoring network designed and implemented, feedback information on compliance is obtained at the end of year I. This information is utilised to obtain a new pumping strategy corresponding to the last two years in the three-year time horizon. In doing so, the management model is updated with the salinity concentration at the end of year I as occurred and monitored in the field.

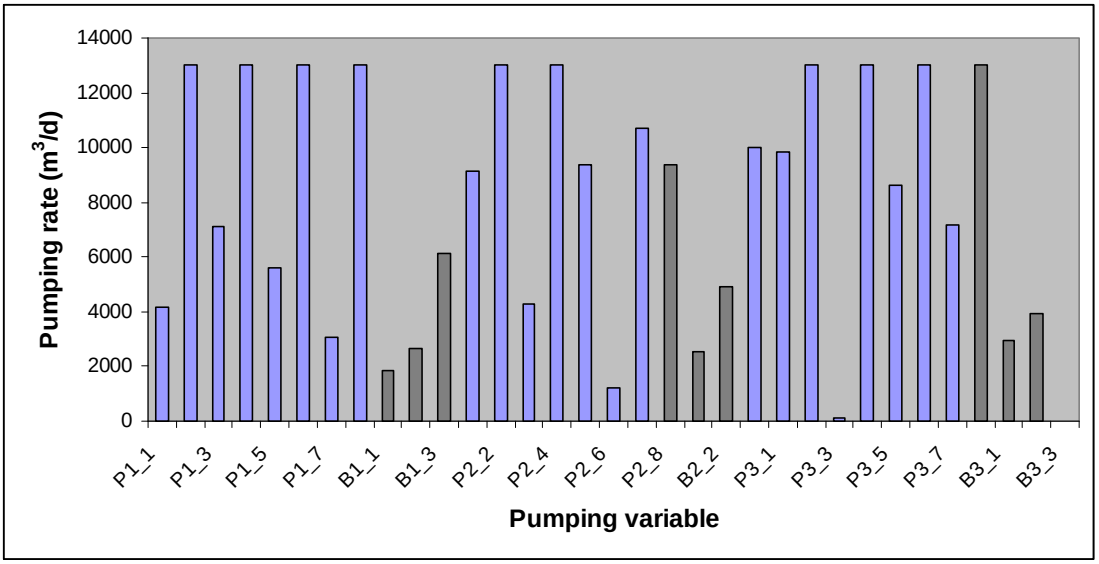

Figure 6.7 Initial optimal pumping rates

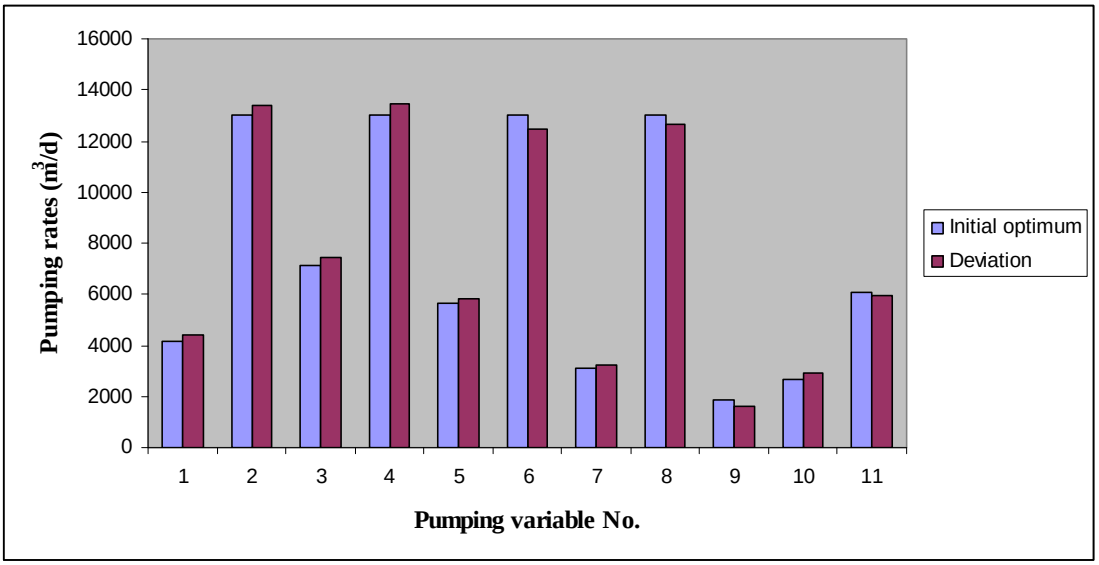

Figure 6.8 Deviation from the optimum values in year I implementation

A deviation in the implementation of the optimal pumping strategy for the first year of operation is illustrated by specifying randomly perturbed pumping values originally obtained as solutions. Deviations in the range 0–20 % are considered. These perturbed pumping values are considered as actual pumping occurring in the field. This is scenario II. An example of deviated pumping rates for the first year in comparison to the actual optimum values is shown in figure 6.8. To address the ill effects of the deviation in the implemented strategies on the concentration levels at the end of the time horizon of operation, the aquifer management strategies for the future time periods are modified based on new strategies derived using simulation-

optimisation. This gives the new optimal rates of pumping for the future time periods considered in the aquifer management. This is scenario III. The objective function value obtained is 217670. It is seen that the objective function value has decreased in comparison with the initially obtained optimum value. The new optimal pumping rates obtained for the second and third years of operation, in comparison with the corresponding values obtained from the initial optimal solution are shown in figure 6.9.

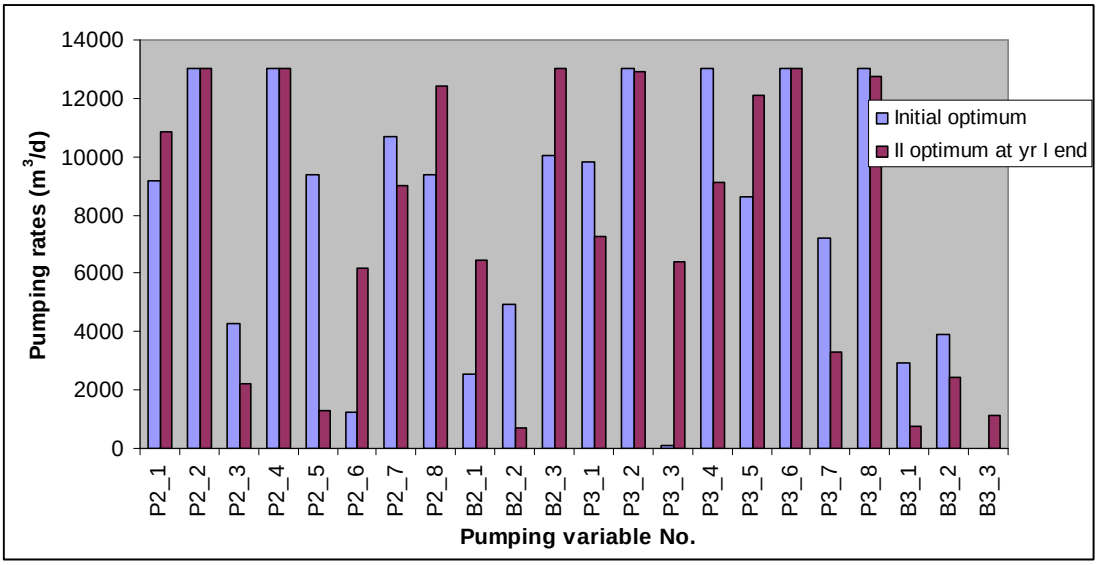

Figure 6.9 Comparison of initial optimum and new optimal pumping rates for years 2 and 3

In a similar manner, random deviations from the newly prescribed optimum were considered for the second year of operation are considered in scenario IV. A random deviation from the prescribed strategy for the second year of operation is shown in figure 6.10 and the new optimal strategies developed for the third year of operation in comparison with the optimum prescribed at the end of first year is shown in figure 6.11. Scenario V depicts the new optimal operation strategies derived for the third year of operation. Scenario VI considers a random deviation in the pumping values obtained for the third year.

The objective function value obtained is 204802 which is considerably less than the initial and second optimum values. It could be noted that as the field implementation of the management strategies deviate from the prescribed values, it results in a decrease in the benefits accrued, which in this study is the total sustainable

rates of pumping. The optimal pumping values for the third year as obtained from the initial optimisation and the two subsequent modifications are shown in figure 6.11.

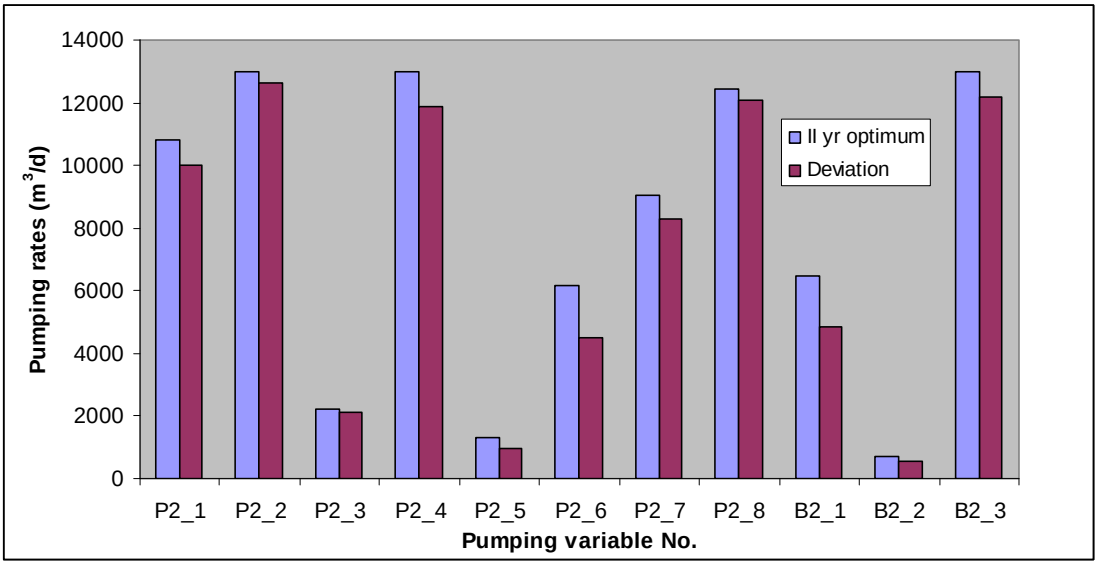

Figure 6.10 Deviation from optimum values in year II implementation

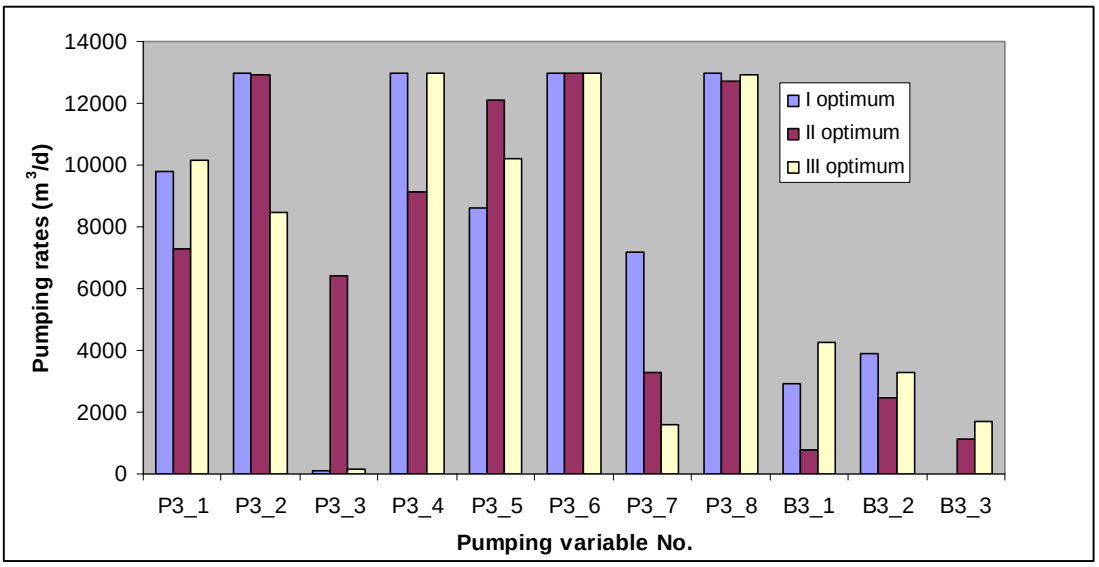

Figure 6.11 Comparison of optimal solutions for third year operation

The objective function values obtained as the initial optimum and the two subsequent modifications for six different random realisations of the deviations in pumping in implementing the optimum strategies are shown in figure 6.12. As evident, the deviations in the implementation led to a decrease in the value of total pumping that could be achieved.

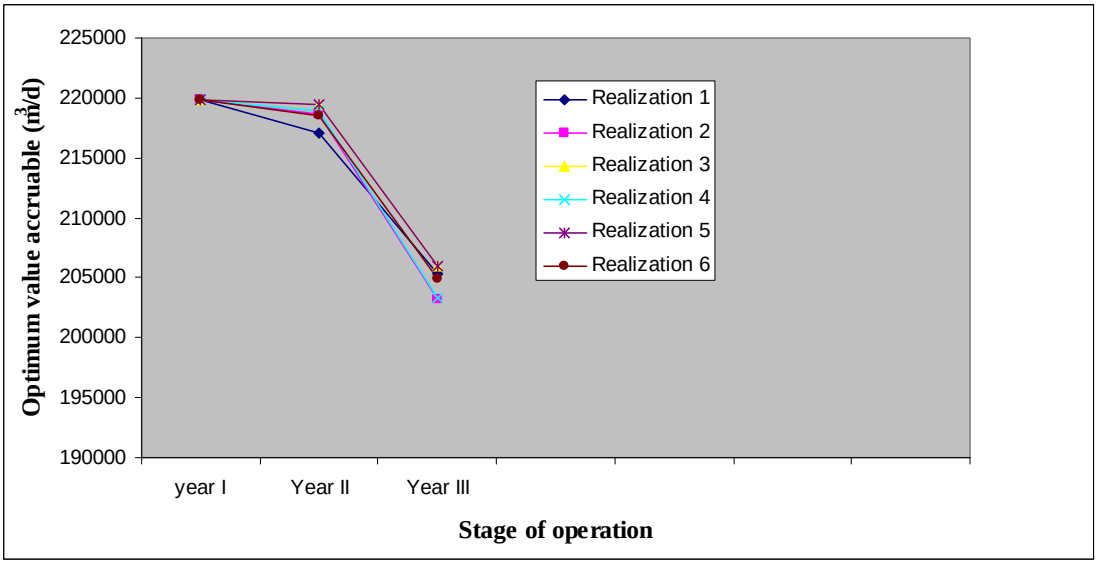

Figure 6.12 Objective function value of total pumping for six realisations of deviations

The concentrations at the 10 monitoring locations for one realisation of each of the six scenarios are shown in table 1. The effect of the deviation in the implemented strategies on the concentration levels at the monitoring locations was quantified as the mean absolute values of the deviation in the concentration values from the expected values. Mean absolute error (MAE) values were computed over 6 different realisations of the deviations. The MAE values of the deviations of concentrations at the monitoring locations for all other scenarios from the concentrations corresponding to scenario I are given in table 2. From the values of mean absolute errors of deviations it is evident that as the implemented strategies deviated from the prescribed ones, the deviation in concentration increased. The MAE value is the highest for all monitoring locations in scenario II. Scenario II considers implementation of the exact optimal strategies for years 2 and 3 in spite of the deviation observed in the first year. Thus, it may be inferred that, given that a deviation occurs in the first time-step of operation, the prescribed strategy may not remain optimal even if the pumping strategies for the subsequent time-steps are exactly implemented. When corrective measures are developed for this deviation in scenario III, it could be observed that concentration levels close to that in scenario I could be achieved, although there was compromise on the objective function value. Similarly, corrective measures to the deviations in the second year are also illustrated. Deviations in concentrations corresponding to scenario VI, which sequentially updates the pumping strategies for each year, is are less than those corresponding to

scenario II. This indicates that sequentially modifying the pumping strategies based on the deviations occurring in the field helps in achieving better compliance.

Table 6.1 Concentrations at 10 observation locations for six scenarios corresponding to a single realisation of deviation in pumping

Monitoring location	Scenario I	Scenario II	Scenario III	Scenario IV	Scenario V	Scenario VI
	Concentrations at observation locations (mg/L)					
1	586.10	541.56	592.08	604.40	605.61	605.72
2	1240.99	1280.07	1239.40	1260.70	1260.64	1261.38
3	2460.89	2431.38	2457.23	2464.56	2463.24	2462.31
4	2316.46	2307.34	2323.55	2322.34	2321.51	2325.31
5	3600.98	3612.13	3603.18	3607.97	3608.01	3612.59
6	3737.35	3719.48	3746.69	3751.11	3752.12	3748.37
7	1163.40	1146.87	1161.09	1176.41	1177.76	1174.66
8	3862.56	3838.25	3856.32	3867.91	3868.79	3865.33
9	2937.99	2960.12	2944.88	2949.38	2949.79	2952.58
10	453.11	465.58	446.93	462.73	462.51	467.21

Table 6.2 Mean absolute error of deviations of other optimal scenarios from scenario I

Monitoring location	Scenario II	Scenario III	Scenario IV	Scenario V	Scenario VI
	MAE of deviations of other scenarios from scenario I				
1	42.98	5.29	16.88	20.53	18.52
2	37.40	2.15	19.78	19.32	21.24
3	30.85	3.43	4.34	2.77	2.57
4	9.11	5.93	5.29	4.35	9.32
5	10.33	2.86	5.42	6.18	11.98
6	17.13	8.73	14.71	15.51	12.98
7	16.36	3.11	13.78	14.45	13.12
8	25.35	7.64	6.13	4.30	2.68
9	22.05	8.26	12.30	10.62	15.00
10	14.66	7.50	9.73	11.13	14.15

Monitoring network designs for groundwater pollution monitoring usually uses a dynamic design in which monitoring wells are dynamically designed for monitoring at different time phases. This is because most of these designs are applied to pollutant plumes which are dynamic, i.e. they move with time. In the present work, the objective of the designed network is to evaluate the compliance of an aquifer management strategy to a prescribed sustainable pumping strategy. The prescribed strategy is developed to control saltwater intrusion into the land and hence the saltwater plume is not expected to be dynamic at least in the short time frames considered for management. Hence, the design of the monitoring networks was based

on the originally specified optimal pumping scenario, for the entire time horizon of aquifer management.

6.5 Conclusions

An optimal pumping strategy for a coastal aquifer was developed using linked simulation-optimisation. For the developed optimal strategy a monitoring network is designed which could be used to evaluate the compliance of the implemented strategies to those prescribed by the optimal solution. A new objective for optimal design of a compliance monitoring network for saltwater intrusion monitoring is developed. The new objective decides whether a potential monitoring location should be chosen based on the uncertainty in the concentration value at that point as well as the correlation between the concentrations of that location and other chosen monitoring locations. Thus it helps to choose monitoring locations at regions of high uncertainty and at the same time helps to reduce redundancy by reducing the spatial correlation between the monitoring locations. A sequential method of determining the pumping strategies for future time periods of operation to nullify the deviations that occurred in the field implementation during the previous time periods is also proposed. It was found that sequential modification of the strategies helps in better compliance with the imposed constraints. This is because the pumping strategies are modified based on the feedback information from the field. This feedback information helps in optimal modification of future management strategies.

7. SUMMARY AND CONCLUSION

7.1 Summary

Simulation-optimisation based methodologies were developed for obtaining optimal groundwater extraction strategies for the management of saltwater intrusion in coastal aquifers. A new surrogate modelling approach based on genetic programming was developed to predict the saltwater intrusion process in coastal aquifers induced by groundwater extraction. Coupled simulation-optimisation approaches using genetic programming and modular neural network based surrogate models and multi-objective genetic algorithm were developed. They were found to be computationally efficient methodologies for obtaining coastal aquifer management strategies. Other features of the newly developed methodology include search space adaptation for further computational efficiency and adaptive training of the surrogate models. Ensemble surrogate model based methodology was proposed to address the predictive uncertainty inherent to surrogate based simulation-optimisation. This methodology was extended to develop coastal aquifer management strategies under parameter uncertainty. A number of surrogate models were trained and tested over different regions of parameter decision space to generate an ensemble of surrogate models. The surrogate models in the ensemble were then coupled to a multi-objective genetic algorithm in a multiple realisation-optimisation framework to obtain stochastic and robust optimal strategies for coastal aquifer management under parameter uncertainty. The methodology was applied to a well field in a realistic coastal aquifer system in the Burdekin region of Queensland in Australia. Compliance monitoring network design with a new objective function was also developed. Compliance monitoring well locations for the considered well field for a specific optimal pumping strategy were obtained using the proposed monitoring network design formulation. Sequential modification of the optimal strategies at different stages of the implementation based on the compliance information from the monitoring network was also illustrated using numerical experiments. The developed methodologies contribute towards integrated management and monitoring for the sustainable management of saltwater intrusion in coastal aquifers.

7.2 Conclusions

The main conclusions derived from this study are as follows;

Simulation-optimization approaches based on genetic programming and modular neural network based surrogate models and multi-objective genetic algorithm were developed. Both the surrogate modeling approaches proved to be useful in achieving computational efficiency in deriving optimal pumping strategies. Simulation-optimization approach developed using genetic programming was found to have definite advantages over modular neural network like, the lesser number of parameters and parsimony in the identification of input variables and advantages in the implementation of search space adaptation.

A new methodology based on ensemble surrogate modeling and multiple realization optimization was developed for improving the reliability of optimal solutions obtained using surrogate based simulation-optimization. The ill-effects of the predictive uncertainty of the surrogate models on the optimal solutions can be reduced using this approach thereby improving the reliability of the optimal solutions. It was illustrated that solutions with a reliability level as high as 0.99 can be obtained using this approach where as a single surrogate model based approach has a reliability level of around 0.5. The multiple realization approach was validated using a chance constrained optimization approach which produced similar results.

The proposed ensemble surrogate modelling with multiple realization optimization approach was extended to solve coastal aquifer management problem under parameter uncertainty. Hydraulic conductivity and annual aquifer recharge were considered as uncertain parameters in modeling 3D density dependent flow and transport using FEMWATER. The ensemble surrogate models were coupled with multi-objective genetic algorithm in a multiple realization optimization framework to derive stochastic optimal solutions. The obtained optimal solutions are robust within the range of uncertain parameters considered in the model development.

A new formulation for the optimal design of compliance monitoring network was developed and applied to a specific optimal pumping strategy developed in this study. In addition to the objective of placing a monitoring well at locations of maximum uncertainty, minimization of the correlation in concentration values between chosen locations of monitoring was incorporated in a single objective optimization. A

formulation which considers only maximization of uncertainty would tend to locate monitoring wells in locations of maximum uncertainty. This may result in the placement of many monitoring wells in small regions there by resulting in redundancy in the information collected. The additional component considered in the new formulation help to avoid the placement of a monitoring well in a location if the concentration at this location has high correlation with another location already chosen for monitoring. As spatial correlation exists in the values of salinity concentrations, this objective tends to spatially disperse the monitoring locations.

The information obtained from the monitoring network design can be used to evaluate the compliance of the implemented strategies to the prescribed ones. If non-compliance is observed, it is necessary to update the optimal strategies to compensate for the deviations occurred in the already implemented stages of management. Updating the optimal pumping strategies after each stage of implementation, based on the compliance information was found to help in achieving better compliance at the end of the time horizon of management.

7.3 Recommendations for future work

Genetic programming was introduced as a potential surrogate modelling tool for simulation-optimisation of groundwater management in this study. Modular neural network and genetic programming based surrogate models were developed for solving the coastal aquifer management problem and were compared. The application of genetic programming based surrogate models may be tested for other complex groundwater management problems like pump-and-treat design. Also, the GP based surrogate modelling approach may be compared to other surrogate modelling approaches to evaluate the advantages and disadvantages. Ensemble-based surrogate modelling with multiple realisation optimisation was introduced as a methodology to solve the coastal aquifer management problem under parameter uncertainty. As a pilot study in this direction, the methodology was tested for homogenous and uniform values of the uncertain parameters, hydraulic conductivity and recharge. This may be extended to heterogeneous and varying fields of parameter values in future studies.

REFERENCES

Abarca, E, Vazquez-Sune, E, Carrera, J, Capino, B, Gamez, D & Batlle, F 2006, 'Optimal design of measures to correct seawater intrusion', *Water Resources Research,* vol. 42.

Abd-Elhamid, HF & Javadi, AA 2011, 'A Cost-Effective Method to Control Seawater Intrusion in Coastal Aquifers', *Water Resources Management,* vol. 25**,** pp. 2755–2780.

Australian Bureau of Statistics 2004, *Regional Population Growth, Australia and New Zealand**,** 2001–02,* cat. no. 3218.0.

Ahlfeld, DP & Pinder, GF 1992, 'A Fast And Accurate Method For Solving Subsurface Contaminant Transport Problems With A Single Uncertain Parameter', *Advances in Water Resources,* vol. 15**,** pp. 143–150.

Alcolea, A, Renard, P, Mariethoz, G & Bertone, F 2009. Reducing the impact of a desalination plant using stochastic modeling and optimization techniques. *Journal of Hydrology,* vol. 365**,** pp. 275–288.

Aly, AH & Peralta, RC 1999, 'Optimal design of aquifer cleanup systems under uncertainty using a neural network and a genetic algorithm. *Water Resources Research,* vol. 35**,** 2523–2532.

Ammar, K, Khalil, A, Mckee, M & Kaluarachchi, J 2008, 'Bayesian deduction for redundancy detection in groundwater quality monitoring networks', *Water Resources Research,* vol. 44.

Arndt, O, Barth, T, Freisleben, B & Grauer, M 2005, 'Approximating a finite element model by neural network prediction for facility optimization in groundwater engineering', *European Journal of Operational Research,* vol. 166**,** pp. 769–781.

Asefa, T, Kemblowski, M, Lall, U & Urroz, G 2005, 'Support vector machines for nonlinear state space reconstruction: Application to the Great Salt Lake time series', *Water Resources Research,* vol. 41.

Asefa, T, Kemblowski, MW, Urroz, G, Mckee, M & Khalil, A 2004, 'Support vectors-based groundwater head observation networks design', *Water Resources Research,* vol. 40.

Ataie-Ashtiani, B & Ketabchi, H 2011, 'Elitist Continuous Ant Colony Optimization

Algorithm for Optimal Management of Coastal Aquifers', *Water Resources Management*, vol. 25**,** pp. 165–190.

Ayvaz, MT & Karahan, H 2008, 'A simulation/optimization model for the identification of unknown groundwater well locations and pumping rates,' *Journal of Hydrology*, vol. 357**,** pp. 76–92.

Babovic, V & Keijzer, M 2002, 'Rainfall runoff modelling based on genetic programming', *Nordic Hydrology*, vol. 33**,** pp. 331–346.

Back, W & Freeze, RA 1983, 'Chemical Hydrogeology', *Benchmark papers in Geology*, 73, Hutchinson Ross Publications, Co., Stroudsburg, PA.

Ball J, Donnelley L, Erlanger, P, Evans, R, Kollmorgen, A, Neal, B & Shirley, M 2001, *Inland waters, Australia State of the Environment Report 2001*, CSIRO Publishing, Canberra, Australia.

Barlow, PM & Reichard, EG 2010, 'Saltwater intrusion in coastal regions of North America', *Hydrogeology Journal*, vol. 18**,** pp. 247–260.

Bashi-Azghadi, SN & Kerachian, R 2010. Locating monitoring wells in groundwater systems using embedded optimization and simulation models. *Science of the Total Environment*, vol. 408**,** pp. 2189–2198.

Bau, DA & Mayer, AS 2008. Optimal design of pump-and-treat systems under uncertain hydraulic conductivity and plume distribution. *Journal of Contaminant Hydrology*, vol. 100**,** pp. 30–46.

Bayer, P, Buerger, CM & Finkel, M 2008, 'Computationally efficient stochastic optimization using multiple realizations', *Advances in Water Resources*, vol. 31**,** pp. 399–417.

Bear, J, Cheng, AH-D, Sorek, S, Ouazar, D & Herrera, I 1999, *Seawater intrusion in coastal aquifers–concepts, methods and practices*, Kluwer Academic Publishers, London.

Bear, J & Dagan, G 1964, 'Some exact solutions of interface problems by means of the hodograph method', *Journal of Geophysical Research*, vol. 69, no. 2, pp. 1563–1572.

Behzadian, K, Kapelan, Z, Savic, D & Ardeshir, A 2009, 'Stochastic sampling design using a multi-objective genetic algorithm and adaptive neural networks', *Environmental Modelling & Software*, vol. 24**,** pp. 530–541.

Bhattacharjya, R & Datta, B 2005, 'Optimal management of coastal aquifers using linked simulation optimization approach', *Water Resources Management*, vol.

19**,** pp. 295–320.

Bhattacharjya, RK & Datta, B 2009, 'ANN-GA-Based Model for Multiple Objective Management of Coastal Aquifers', *Journal of Water Resources Planning and Management–ASCE,* vol. 135**,** pp. 314–322.

Bhattacharjya, RK, Datta, B & Satish, MG 2007, 'Artificial neural networks approximation of density dependent saltwater intrusion process in coastal aquifers', *Journal of Hydrologic Engineering,* vol. 12**,** pp. 273–282.

Bierkens, MFP, Knotters, M & Hoogland, T 2001, 'Space-time modeling of water table depth using a regionalized time series model and the Kalman filter', *Water Resources Research,* vol. 37**,** pp. 1277–1290.

Blair, PM & Turner, N 2004, *Groundwater—a crucial element of water recycling in Perth, Western Australia*, WSUD, Sydney Metropolitan Catchment Management Authority, Sydney, NSW, Australia, viewed 2 September 2008, http://www.wsud.org/literature.htm.

Brown, JS 1925, 'Study of Coastal groundwater with special reference to Connecticut', *U.S. Geological Survey Water-Supply Paper*, 537.

Brown, CJ & Misut, PE 2010, 'Aquifer geochemistry at potential aquifer storage and recovery sites in coastal plain aquifers in the New York City area, USA', *Applied Geochemistry,* vol. 25**,** pp. 1431–1452.

Chadalavada, S & Datta, B 2008, 'Dynamic optimal monitoring network design for transient transport of pollutants in groundwater aquifers', *Water Resources Management,* vol. 22**,** pp. 651–670.

Chadalavada, S, Datta, B & Naidu, R 2011, 'Uncertainty based optimal monitoring network design for a chlorinated hydrocarbon contaminated site', *Environmental Monitoring and Assessment,* vol. 173**,** pp. 929–940.

Chan, N 1993, 'Robustness of the multiple realization method for stochastic hydraulic aquifer management', *Water Resources Research,* 29**,** 3159–3167.

Cheng, AHD, Halhal, D, Naji, A & Ouazar, D 2000, 'Pumping optimization in saltwater-intruded coastal aquifers', *Water Resources Research,* vol. 36**,** pp. 2155–2165.

Cherubini, C & Pastore, N 2011, 'Critical stress scenarios for a coastal aquifer in southeastern Italy', *Natural Hazards and Earth System Sciences,* vol. 11**,** pp. 1381–1393.

Dagan, G & Zeitoun, DG 1998a, 'Free-surface flow toward a well and interface

upconing in stratified aquifers of random conductivity', *Water Resources Research,* vol. 34**,** pp. 3191–3196.

Dagan, G & Zeitoun, DG 1998b, 'Seawater-freshwater interface in a stratified aquifer of random permeability distribution', *Journal of Contaminant Hydrology,* vol. 29**,** 185–203.

Darnault, CJG & Godinez, IG 2008, 'Coastal aquifers and saltwater intrusion', *Overexploitation and Contamination of Shared Groundwater Resources: Management, (Bio) Technological, and Political Approaches to Avoid Conflicts***,** pp. 185–201.

Das, A & Datta, B 1999a, 'Development of management models for sustainable use of coastal aquifers', *Journal of Irrigation and Drainage Engineering–ASCE,* vol. 125**,** pp. 112–121.

Das, A & Datta, B 1999b, 'Development of multiobjective management models for coastal aquifers', *Journal of Water Resources Planning and Management–ASCE,* vol. 125**,** pp. 76–87.

Das, A & Datta, B 2000, 'Optimization based solution of density dependent seawater intrusion in coastal aquifers', *Journal of Hydrologic Engineering,* vol. 5**,** pp. 82–89.

Datta, B, Chakrabarty, D & Dhar, A 2009a, 'Simultaneous identification of unknown groundwater pollution sources and estimation of aquifer parameters', *Journal of Hydrology,* vol. 376**,** pp. 48–57.

Datta, B & Dhiman, SD 1996, 'Chance-constrained optimal monitoring network design for pollutants in ground water', *Journal of Water Resources Planning and Management–ASCE,* vol. 122**,** pp. 180–188.

Datta, B, Vennalakanti, H & Dhar, A 2009, 'Modeling and control of saltwater intrusion in a coastal aquifer of Andhra Pradesh, India', *Journal of Hydro-Environment Research,* vol. 3**,** pp. 148–158.

Deb, K 2001, *Multi-objective optimization using evolutionary algorithms*, John Wiley and Sons, Ltd.

Dhar, A & Datta, B 2007, 'Multiobjective design of dynamic monitoring networks for detection of groundwater pollution', *Journal of Water Resources Planning and Management–ASCE,* vol. 133**,** pp. 329–338.

Dhar, A & Datta, B 2009a, 'Saltwater Intrusion Management of Coastal Aquifers. I: Linked Simulation-Optimization', *Journal of Hydrologic Engineering,* vol. 14**,**

pp. 1263–1272.

Dhar, A & Datta, B 2009b, 'Saltwater Intrusion Management of Coastal Aquifers. II: Operation Uncertainty and Monitoring', *Journal of Hydrologic Engineering,* vol. 14**,** pp. 1273–1282.

Dhar, A & Datta, B 2009c, 'Global Optimal design of Ground Water Monitoring Network Using Embedded Kriging', *Ground Water,* vol. 47**,** pp. 806–815.

Diamantopoulou, P. & Voudouris, K. 2008, 'Optimization of water resources management using SWOT analysis: The case of Zakynthos Island, Ionian Sea, Greece', *Environmental Geology,* vol. 54**,** pp. 197–211.

Diersch, H-JG 2002, *FEFLOW finite element subsurface flow and transport simulation system reference manual,* WASY Inst. For Water Resources Planning and System Research, Berlin.

Dokou, Z & Pinder, GF 2009, 'Optimal search strategy for the definition of a DNAPL source', *Journal of Hydrology,* vol. 376**,** pp. 542–556.

Dokou, Z & Pinder, GF 2011, 'Extension and field application of an integrated DNAPL source identification algorithm that utilizes stochastic modeling and a Kalman filter', *Journal of Hydrology,* vol. 398**,** pp. 277–291.

Dorado, J, Rabunal, JR, Pazos, A, Rivero, D, Santos, A & Puertas, J 2003, 'Prediction and modeling of the rainfall-runoff transformation of a typical urban basin using ANN and GP', *Applied Artificial Intelligence,* vol. 17**,** pp. 329–343.

Emch, PG & Yeh, WWG 1998, 'Management model for conjunctive use of coastal surface water and ground water', *Journal of Water Resources Planning and Management–ASCE,* vol. 124**,** pp. 129–139.

Feyen, L & Gorelick, SM 2004, 'Reliable groundwater management in hydroecologically sensitive areas', *Water Resources Research,* vol. 40.

Feyen, L & Gorelick, SM 2005, 'Framework to evaluate the worth of hydraulic conductivity data for optimal groundwater resources management in ecologically sensitive areas', *Water Resources Research,* vol. 41**,** 13.

Finney, BA, Samsuhadi & Willis, R 1992, 'Quasi-3-dimensional optimization model of Jakarta basin', *Journal of Water Resources Planning and Management–ASCE,* vol. 118**,** pp. 18–31.

Francone, FD 1998, *Discipulus™ Software Owner's Manual, version 3.0 DRAFT,* Machine Learning Technologies Inc., Littleton, CO, USA.

Garson, GD 1991, 'Intelligence neural network connection weights', *Artificial*

Intelligence Expert, vol. 6, pp. 47–51

Gaur, S & Deo, MC 2008, 'Real-time wave forecasting using genetic programming', *Ocean Engineering*, vol. 35**,** pp. 1166–1172.

Gevrey, M, Dimopoulos, L & Lek, S 2003, 'Review and comparison of methods to study the contribution of variables in artificial neural network models', *Ecological Modelling*, vol. 160**,** pp. 249–264.

Giambastiani, BMS, Antonellini, M, Essink, GHPO & Stuurman, RJ 2007a, 'Saltwater intrusion in the unconfined coastal aquifer of Ravenna (Italy): A numerical model', *Journal of Hydrology*, vol. 340**,** pp. 91–104.

Goh, ATC 1995, 'Backpropagation neural networks for modeling complex systems', *Artificial Intelligence in Engineering*, vol. 9**,** pp. 143–151.

Gorelick, SM 1983, 'A review of distributed parameter groundwater-management modeling methods', *Water Resources Research*, vol. 19**,** pp. 305–319.

Gorelick, SM, Voss, CI, Gill, PE, Murray, W, Saunders, MA & Wright, MH 1984, 'Aquifer reclamation design – the use of contaminant transport simulation combined with nonlinear-programming', *Water Resources Research*, vol. 20**,** pp. 415–427.

Graham, W & McLaughlin, D 1989a, 'Stochastic-analysis of nonstationary subsurface solute transport .1. Unconditional moments', *Water Resources Research*, vol. 25**,** pp. 215–232.

Graham, W & McLaughlin, D 1989b, 'Stochastic-analysis of nonstationary subsurface solute transport .2. Conditional moments', *Water Resources Research*, vol. 25**,** pp. 2331–2355.

Guo, W & Langevin, CD 2002, 'User's guide to SEAWAT: a computer program for simulation of three-dimensional variable-density groundwater flow', *US Geological Survey Open File Report 01-434*, USGS.

Guvanasen, V, Wade, SC & Barcelo, MD 2000a, 'Simulation of regional ground water flow and salt water intrusion in Hernando County, Florida', *Ground Water*, vol. 38**,** pp. 772–783.

Haddad, OB & Marino, MA 2011, 'Optimum operation of wells in coastal aquifers', *Proceedings of the Institution of Civil Engineers–Water Management*, vol. 164**,** pp. 135–146.

Hallaji, K & Yazicigil, H 1996, 'Optimal management of a coastal aquifer in southern Turkey', *Journal of Water Resources Planning and Management–ASCE*, vol.

122**,** pp. 233–244.

Hantush, S 1968, 'Unsteady movement of freshwater in thick unconfined saline aquifers', *Bull Int Assoc Sci Hydrol,* 1 vol. 3, pp. 40–60.

He, L, Huang, GH, Lu, HW & Zeng, G-M 2008, 'Optimization of surfactant-enhanced aquifer remediation for a laboratory BTEX system under parameter uncertainty', *Environmental Science & Technology,* vol. 42**,** pp. 2009–2014.

He, L, Huang, GH, Lu, HW 2009, 'A coupled simulation-optimization approach for groundwater remediation design under uncertainty: An application to a petroleum-contaminated site', *Environmental Pollution,* vol. 157**,** pp. 2485–2492.

He, L, Huang, GH, Lu, HW 2010a, 'A stochastic optimization model under modeling uncertainty and parameter certainty for groundwater remediation design–Part I. Model development', *Journal of Hazardous Materials,* vol. 176**,** pp. 521–526.

He, L, Huang, GH, Lu, HW 2010b, 'A stochastic optimization model under modeling uncertainty and parameter certainty for groundwater remediation design: Part II. Model application', *Journal of Hazardous Materials,* vol. 176**,** pp. 527–534.

Henry, HR 1959, 'Salt intrusion into freshwater aquifers', *Journal of Geophysical Research,* vol. 64, pp. 1911–1919.

Herrera, GS & Pinder, GF 1998, 'Cost-effective groundwater quality sampling network design', *Computational Methods in Contamination and Remediation of Water Resources: Proceedings of 12th International Conference on Computational Methods in Water Resources,* vol. 1, 12**,** 51–58.

Herrera, GS & Pinder, GF 2005, 'Space-time optimization of groundwater quality sampling networks', *Water Resources Research,* vol. 41.

Holzbecher, E 1998, *Modelling density-driven flow in porous media,* Springer, Berlin.

Hudak, PF & Loaiciga, HA 1992, 'A location modeling approach for groundwater monitoring network augmentation', *Water Resources Research,* vol. 28**,** pp. 643–649.

Hudak, PF & Loaiciga, HA 1993, 'An optimization method for monitoring network design in multilayered groundwater-flow systems', *Water Resources Research,* vol. 29**,** pp. 2835–2845.

Hudak, PF, Loaiciga, HA & Marino, MA 1995, 'Regional-scale ground water quality

monitoring via integer programming', *Journal of Hydrology (Amsterdam)*, vol. 164**,** pp. 153–170.

HydroGeoLogic Inc. 2002, *MODHMS-MODFLOW based hydrologic modeling system: documentation and user's guide*, Herndon, Virginia.

Iman, RL & Conover, WJ 1982, 'A distribution-free approach to inducing rank correlation among input variables', *Communications in Statistics Part B–Simulation and Computation*, vol. 11**,** pp. 311–334.

Iribar, V, Carrera, J, Custodio, E & Medina, A 1997, 'Inverse modelling of seawater intrusion in the Llobregat delta deep aquifer', *Journal of Hydrology*, vol. 198**,** pp. 226–244.

Jin, Y 2005, 'A comprehensive survey of fitness approximation in evolutionary computation', *Soft Computing*, vol. 9**,** pp. 3–12.

Karterakis, SM, Karatzas, GP, Nikolos, IK & Papadopoulou, MP 2007, 'Application of linear programming and differential evolutionary optimization methodologies for the solution of coastal subsurface water management problems subject to environmental criteria', *Journal of Hydrology*, vol. 342**,** pp. 270–282.

Katsifarakis, KL & Petala, Z 2006, 'Combining genetic algorithms and boundary elements to optimize coastal aquifers' management', *Journal of Hydrology*, vol. 327**,** pp. 200–207.

Kentel, E, Gill, H & Aral, MM 2005, 'Evaluation of groundwater potential of Savannah Georgia region', *Multimedia Environmental Simulations Laboratory Report No. MESL-01-05*, MESL.

Kipp, KL 1986, 'HST3D, a computer code for simulation of heat and solute transport in three-dimensional groundwater flow systems', *US Geological Survey Water Resources Investigations Report 86-4095*, USGS.

Kipp, KL 1997, 'Guide to the revised heat and solute transport simulator, HST3D-Version 2', *Water Resources Investigation Report 97-4157*, USGS.

Ko, N-Y & Lee, K-K 2009, 'Convergence of deterministic and stochastic approaches in optimal remediation design of a contaminated aquifer', *Stochastic Environmental Research and Risk Assessment*, vol. 23**,** pp. 309–318.

Kollat, JB & Reed, PM 2007, 'A computational scaling analysis of multiobjective evolutionary algorithms in long-term groundwater monitoring applications', *Advances in Water Resources*, vol. 30**,** pp. 408–419.

Kollat, JB & Reed, PM & Kasprzyk, JR 2008, 'A new epsilon-dominance hierarchical Bayesian optimization algorithm for large multiobjective monitoring network design problems', *Advances in Water Resources,* vol. 31**,** pp. 828–845.

Kollat, JB & Reed, PM & Maxwell, RM 2011, 'Many-objective groundwater monitoring network design using bias-aware ensemble Kalman filtering, evolutionary optimization, and visual analytics', *Water Resources Research,* vol. 47.

Kourakos, G & Mantoglou, A 2008, 'Remediation of heterogeneous aquifers based on multiobjective optimization and adaptive determination of critical realizations', *Water Resources Research,* vol. 44.

Kourakos, G & Mantoglou, A 2009, 'Pumping optimization of coastal aquifers based on evolutionary algorithms and surrogate modular neural network models', *Advances in Water Resources,* vol. 32**,** pp. 507–521.

Kourakos, G & Mantoglou, A 2011, 'Simulation and Multi-Objective Management of Coastal Aquifers in Semi-Arid Regions', *Water Resources Management,* vol. 25**,** pp. 1063–1074.

Koza, JR 1994, 'Genetic programming as a means for programming computers by natural-selection', *Statistics and Computing,* vol. 4**,** pp. 87–112.

Lin, H-CJ, Richards, DR, Talbot, CA, Yeh, G-T, Cheng, J-R, Cheng, H-P & Jones, NL 1997, *A three-dimensional finite element computer model for simulating density-dependent flow and transport in variable saturated media: Version 3.0,* U.S Army Engineer Research and Development Center, Vicksburg, Miss. USA.

Lin, J, Snodsmith, JB, Zheng, CM & Wu, JF 2009, 'A modeling study of seawater intrusion in Alabama Gulf Coast, USA', *Environmental Geology,* vol. 57**,** pp. 119–130.

LINDO Systems Inc. 1999, *LINGO User's Guide,* Chicago.

Liu, F, Anh, VV, Turner, I, Bajracharya, K, Huxley, WJ & Su, N 2006, 'A finite volume simulation model for saturated-unsaturated flow and application to Gooburrum, Bundaberg, Queensland, Australia', *Applied Mathematical Modelling,* vol. 30**,** pp. 352–366.

Loaiciga, HA, Charbeneau, RJ, Everett, LG, Fogg, GE, Hobbs, BF & Rouhani, S 1992, 'Review of groundwater quality monitoring network design', *Journal of Hydraulic Engineering–ASCE,* vol. 118**,** pp. 11–37.

Mahar, PS & Datta, B 1997, 'Optimal monitoring network and ground-water-pollution source identification', *Journal of Water Resources Planning and Management–ASCE,* vol. 123**,** pp. 199–207.

Maimone, M 2002, 'Developing an effective coastal aquifer management program', *Proceedings of the 17th Salt Water Intrusion Meeting, Delft, The Netherlands*, pp. 327–336.

Makkeasorn, A, Chang, NB & Zhou, X 2008, 'Short-term streamflow forecasting with global climate change implications – A comparative study between genetic programming and neural network models', *Journal of Hydrology,* pp. 352**,** 336–354.

Mantoglou, A 2003, 'Pumping management of coastal aquifers using analytical models of saltwater intrusion', *Water Resources Research,* vol. 39**,** 12.

Mantoglou, A & Papantoniou, M 2008, 'Optimal design of pumping networks in coastal aquifers using sharp interface models', *Journal of Hydrology,* vol. 361**,** pp. 52–63.

Mantoglou, A, Papantoniou, M & Giannoulopoulos, P 2004, 'Management of coastal aquifers based on nonlinear optimization and evolutionary algorithms', *Journal of Hydrology*, vol. 297**,** pp. 209–228.

Martin, RR 1997, 'Sustainability of supplies from a coastal aquifer and the impact of artificial recharge: Lefevre Peninsula, South Australia', Masters Thesis, Flinders University, Adelaide. Australia.

Masoumi, F & Kerachian, R 2010, 'Optimal redesign of groundwater quality monitoring networks: a case study', *Environmental Monitoring and Assessment,* vol. 161**,** pp. 247–257.

Massmann, J & Freeze, RA 1987a, 'Groundwater contamination from waste management sites – the interaction between risk-based engineering design and regulatory policy 1. Methodology', *Water Resources Research,* vol. 23**,** pp. 351–367.

Massmann, J & Freeze, RA 1987b, 'Groundwater contamination from waste management sites – the interaction between risk-based engineering design and regulatory policy 2. Results', *Water Resources Research,* vol. 23**,** pp. 368–380.

McKinney, DC & Loucks, DP 1992, 'Network design for predicting groundwater contamination', *Water Resources Research,* vol. 28**,** pp. 133–147.

G.A. McMahon, N.J. Arunakumaren, K. Bajracharya (2000) Hydrogeological conceptualization of the Burdekin River Delta Hydro 2000—Proceedings of the 3rd International Hydrology and Water Resources Symposium of the Institution of Engineers, Perth, Western Australia, Australia

McPhee, J & Yeh, WWG 2006, 'Experimental design for groundwater modeling and management', *Water Resources Research,* vol. 42**,** 13.

Melloul, AJ & Goldenberg, LC 1997, 'Monitoring of seawater intrusion in coastal aquifers: Basics and local concerns', *Journal of Environmental Management,* vol. 51**,** pp. 73–86.

Meyer, PD & Brill, ED 1988, 'A method for locating wells in a groundwater monitoring network under conditions of uncertainty', *Water Resources Research,* vol. 24**,** pp. 1277–1282.

Meyer, PD, Valocchi, AJ, Ashby, SF & Saylor, PE 1989, 'A numerical investigation of the conjugate-gradient method as applied to 3-dimensional groundwater-flow problems in randomly heterogeneous porous-media', *Water Resources Research,* vol. 25**,** pp. 1440–1446.

Misut, PE & Voss, CI 2004, 'Simulation of seawater intrusion resulting from proposed expanded pumpage in New York City, USA', in CT Miller, MW Farthing, WG Gray and G Pinders (eds) *Computational Methods in Water Resources, Vols 1 and 2.*

Misut, PE & Voss, CI 2007, 'Freshwater-saltwater transition zone movement during aquifer storage and recovery cycles in Brooklyn and Queens, New York City, USA', *Journal of Hydrology,* vol. 337**,** pp. 87–103.

Montas, HJ, Mohtar, RH, Hassan, AE & Alkhal, FA 2000, 'Heuristic space-time design of monitoring wells for contaminant plume characterization in stochastic flow fields', *Journal of Contaminant Hydrology,* vol. 43**,** pp. 271–301.

Morgan, DR, Eheart, JW & Valocchi, AJ 1993, 'Aquifer remediation design under uncertainty using a new chance constrained programming technique', *Water Resources Research,* vol. 29**,** pp. 551–561.

Naji, A, Cheng, ABD & Ouazar, D 1999, 'BEM solution of stochastic seawater intrusion problems', *Engineering Analysis with Boundary Elements,* vol. 23**,** pp. 529–537.

Narayan, KA, Schleeberger, C & Bristow, KL 2007, 'Modelling seawater intrusion in

the Burdekin Delta Irrigation Area, North Queensland, Australia', *Agricultural Water Management,* vol. 89**,** pp. 217–228.

Nation, E, Werner, AD & Habermehl, MA 2008, *Australia's coastal aquifers and sea level rise, Science for decision makers brief,* Department of Agriculture Fisheries & Forestry, Bureau of Rural Sciences, Canberra, Australia.

Nikolos, IK, Stergiadi, M, Papadopoulou, MP & Karatzas, GP 2008, 'Artificial neural networks as an alternative approach to groundwater numerical modelling and environmental design', *Hydrological Processes,* vol. 22**,** pp. 3337–3348.

Nunes, LM, Cunha, MC & Ribeiro, L 2004a, 'Groundwater monitoring network optimization with redundancy reduction', *Journal of Water Resources Planning and Management–ASCE,* vol. 130**,** pp. 33–43.

Nunes, LM, Cunha, MC & Ribeiro, L 2004b, 'Optimal space-time coverage and exploration costs in groundwater monitoring networks', *Environmental Monitoring and Assessment,* vol. 93**,** pp. 103–124.

Nunes, LM, Paralta, E, Cunha, MC & Ribeiro, L 2004c, 'Groundwater nitrate monitoring network optimization with missing data', *Water Resources Research,* vol. 40.

Olden, JD & Jackson, DA 2002, 'Illuminating the "black box": a randomization approach for understanding variable contributions in artificial neural networks', *Ecological Modelling,* vol. 154**,** pp. 135–150.

Papadopoulou, MP 2011, 'Optimization approaches for the control of seawater intrusion and its environmental impacts in coastal environment', *Pacific Journal of Optimization,* vol. 7**,** pp. 479–502.

Papadopoulou, MP, Nikolos, IK & Karatzas, GP 2010, 'Computational benefits using artificial intelligent methodologies for the solution of an environmental design problem: saltwater intrusion', *Water Science and Technology,* vol. 62**,** pp. 1479–1490.

Parasuraman, K & Elshorbagy, A 2008, 'Toward improving the reliability of hydrologic prediction: Model structure uncertainty and its quantification using ensemble-based genetic programming framework', *Water Resources Research,* vol. 44**,** 12.

Park, CH & Aral, MM 2004, 'Multi-objective optimization of pumping rates and well placement in coastal aquifers', *Journal of Hydrology,* vol. 290**,** pp. 80–99.

Park, N, Cui, L & Shi, L 2009, 'Analytical Design Curves to Maximize Pumping or

Minimize Injection in Coastal Aquifers', *Ground Water,* vol. 47**,** pp. 797–805.

Parker, J, Kim, U, Kitanidis, PK, Cardiff, M & Liu, XY 2010, 'Stochastic Cost Optimization of Multistrategy DNAPL Site Remediation', *Ground Water Monitoring and Remediation,* vol. 30**,** pp. 65–78.

Perera, EDP, Jinno, K, Tsutsumi, A & Hiroshiro, Y 2010, 'A numerical study of saline contamination of a coastal aquifer', *Proceedings of the Institution of Civil Engineers–Water Management,* vol. 163**,** pp. 367–375.

Petalas, C, Pisinaras, V, Gemitzi, A, Tsihrintzis, VA & Ouzounis, K 2009, 'Current conditions of saltwater intrusion in the coastal Rhodope aquifer system, northeastern Greece', *Desalination,* vol. 237**,** pp. 22–41.

Qahman, K, Larabi, A, Ouazar, D, Naji, A & Cheng, AHD 2005, 'Optimal and sustainable extraction of groundwater in coastal aquifers', *Stochastic Environmental Research and Risk Assessment,* vol. 19**,** pp. 99–110.

Qi, SZ. & Qiu, QL 2011, 'Environmental hazard from saltwater intrusion in the Laizhou Gulf, Shandong Province of China', *Natural Hazards,* vol. 56**,** pp. 563–566.

Qin, XS & Huang, GH 2009, 'Characterizing Uncertainties Associated with Contaminant Transport Modeling through a Coupled Fuzzy-Stochastic Approach', *Water Air and Soil Pollution,* vol. 197**,** pp. 331–348.

Ranjithan, S, Eheart, JW & Garrett, JH 1993, 'Neural network based screening for groundwater reclamation under uncertainty', *Water Resources Research,* vol. 29**,** pp. 563–574.

Rao, SVN, Bhallamudi, SM, Thandaveswara, BS & Mishra, GC 2004a, 'Conjunctive use of surface and groundwater for coastal and deltaic systems', *Journal of Water Resources Planning and Management–ASCE,* vol. 130**,** pp. 255–267.

Rao, SVN, Kumar, S, Shekhar, S & Chakraborty, D 2006, 'Optimal pumping from skimming wells', *Journal of Hydrologic Engineering,* vol. 11**,** pp. 464–471.

Rao, SVN, Kumar, S, Shekhar, S, Sinha, SK & Manju, S 2007a, 'Optimal pumping from skimming wells from the Yamuna River flood plain in north India', *Hydrogeology Journal,* vol. 15**,** pp. 1157–1167.

Rao, SVN & Manju, S 2007a, 'Optimal pumping locations of skimming wells', *Hydrological Sciences Journal–Journal Des Sciences Hydrologiques,* vol. 52**,** pp. 352–361.

Rao, SVN, Sreenivasulu, V, Bhallamudi, SM, Thandaveswara, BS & Sudheer, KP

2004b, ‘Planning groundwater development in coastal aquifers’, *Hydrological Sciences Journal–Journal Des Sciences Hydrologiques,* vol. 49**,** pp. 155–170.

Rao, SVN, Thandaveswara, BS, Bhallamudi, SM & Srivivasulu, V 2003, ‘Optimal groundwater management in deltaic regions using simulated annealing and neural networks’, *Water Resources Management,* vol. 17**,** pp. 409–428.

Reed, P, Minsker, B & Valocchi, AJ 2000, ‘Cost-effective long-term groundwater monitoring design using a genetic algorithm and global mass interpolation’, *Water Resources Research,* vol. 36**,** pp. 3731–3741.

Reed, P, Minsker, BS & Goldberg, DE 2003, ‘Simplifying multiobjective optimization: An automated design methodology for the nondominated sorted genetic algorithm-II’, *Water Resources Research,* vol. 39.

Reed, PM & Minsker, BS 2004, ‘Striking the balance: Long-term groundwater monitoring design for conflicting objectives’, *Journal of Water Resources Planning and Management–ASCE,* vol. 130**,** pp. 140–149.

Reichard, EG & Johnson, TA 2005a, ‘Assessment of regional management strategies for controlling seawater intrusion’, *Journal of Water Resources Planning and Management–ASCW,* vol. 131**,** pp. 280–291.

Reilly, TE & Goodman, AS 1985, ‘Quantitative-analysis of saltwater fresh-water relationships in groundwater systems – a historical-perspective’, *Journal of Hydrology,* vol. 80**,** pp. 125–160.

Rejani, R, Jha, MK & Panda, SN 2009, ‘Simulation-Optimization Modelling for Sustainable Groundwater Management in a Coastal Basin of Orissa, India’, *Water Resources Management,* vol. 23**,** pp. 235–263.

Rogers, LL, Dowla, FU & Johnson, VM 1995, ‘Optimal field-scale groundwater remediation using neural networks and the genetic algorithm’, *Environmental Science & Technology,* vol. 29**,** pp. 1145–1155.

Rouhani, S 1985, ‘Variance reduction analysis’, *Water Resources Research,* vol. 21**,** pp. 837–846.

Ruiz-Cardenas, R, Ferreira, MAR & Schmidt, AM 2010, ‘Stochastic search algorithms for optimal design of monitoring networks’, *Environmetrics,* vol. 21**,** pp. 102–112.

Schuetze, N, Grundmann, J & Schmitz, GH 2011, ‘Prospects for Integrated Water Resources Management (IWRM) through the application of simulation-based optimization methods illustrated by the example of agricultural coastal arid

regions in Oman', *Hydrologie Und Wasserbewirtschaftung,* vol. 55**,** pp. 104–115.

Schmorak, S & Mercado, A 1969, 'Upconing of freshwater – seawater interface below pumping wells, field study', *Water Resources Research,* vol. 5, pp. 1290–1311.

Sedki, A & Ouazar, D 2011, 'Simulation-Optimization Modeling for Sustainable Groundwater Development: A Moroccan Coastal Aquifer Case Study', *Water Resources Management,* vol. 25**,** pp. 2855–2875.

Sherif, M, Mohamed, M, Kacimov, A & Shetty, A 2011, 'Assessment of groundwater quality in the northeastern coastal area of UAE as precursor for desalination', *Desalination,* vol. 273**,** pp. 436–446.

Sheta, AF & Mahmoud, A 2001, 'Forecasting using genetic programming', *Proceedings of the 33rd Southeastern Symposium on System Theory,* March 18–20 2001 Athens, Ohio, USA, IEEE, pp. 343–347.

Shi, L, Cui, L, Park, N & Huyakorn, PS 2011, 'Applicability of a sharp-interface model for estimating steady-state salinity at pumping wells-validation against sand tank experiments', *Journal of Contaminant Hydrology,* vol. 124**,** pp. 35–42.

Sikkema, PC & Van Dam, JC 1982, 'Analytical formulas for the shape of the interface in a semi-confined aquifer', *Journal of Hydrology,* vol. 56**,** pp. 201–220.

Singh, A & Minsker, BS 2008, 'Uncertainty-based multiobjective optimization of groundwater remediation design', *Water Resources Research,* vol. 44.

Sorek, S & Pinder, GF 1999, 'Survey of computer codes and case histories', *Seawater Intrusion in Coastal Aquifers – Concepts, Methods and Practices,* vol. 14**,** pp. 399–461.

Sreekanth, J & Datta, B 2010, 'Multi-objective management of saltwater intrusion in coastal aquifers using genetic programming and modular neural network based surrogate models', *Journal of Hydrology,* vol. 393**,** pp. 245–256.

Sreekanth, J & Datta, B 2011a, 'Comparative Evaluation of Genetic Programming and Neural Network as Potential Surrogate Models for Coastal Aquifer Management', *Water Resources Management,* vol. 25**,** pp. 3201–3218.

Sreekanth, J & Datta, B 2011b, 'Coupled simulation-optimization model for coastal aquifer management using genetic programming-based ensemble surrogate

models and multiple-realization optimization', *Water Resources Research,* vol. 47.

Stein, M 1987, 'Large sample properties of simulations using Latin hypercube sampling', *Technometrics,* vol. 29**,** pp. 143–151.

Storck, P, Eheart, JW & Valocchi, AJ 1997, 'A method for the optimal location of monitoring wells for detection of groundwater contamination in three-dimensional heterogenous aquifers', *Water Resources Research,* vol. 33**,** pp. 2081–2088.

Strack, ODL 1976, 'Single-potential solution for regional interface problems in coastal aquifers', *Water Resources Research,* vol. 12**,** pp. 1165–1174.

Teatini, P, Putti, M, Rorai, C, Mazzia, A, Gambolati, G, Tosi, L & Carbognin, L 2010, 'Modeling the saltwater intrusion in the lowlying catchment of the southern Venice Lagoon, Italy', *Management of Natural Resources, Sustainable Development and Ecological Hazards II,* pp. 127**,** 351–362.

Tiedeman, C & Gorelick, SM 1993, 'Analysis of uncertainty in optimal groundwater contaminant capture design', *Water Resources Research,* vol. 29**,** pp. 2139–2153.

US Geological Survey 2010, *Groundwater information, Freshwater saltwater interaction along Atlantic coast,* USGS, http://water.usgs.gov/ogw/gwrp/saltwater/salt.html.

Van Dam, JC & Sikkema, PC 1982a, 'Approximate solution of the problem of the shape of the interface in a semi confined aquifer', *Journal of Hydrology (Amsterdam),* vol. 56**,** pp. 221–238.

Van Geer, FC, Testroet, CBM & Zhou, YX 1991, 'Using Kalman filtering to improve and quantify the uncertainty of numerical groundwater simulations 1. The role of system noise and its calibration', *Water Resources Research,* vol. 27**,** pp. 1987–1994.

Voss, CI 1984, 'A finite-element simulation model for saturated-unsaturated, fluid-density-dependent groundwater flow with energy transport or chemically-reactive single species solute transport', *US Geological Survey Water–Resources Investigations Report 84–4369.*

Voss, CI & Provost, AM 2002, 'SUTRA, A model for saturated-unsaturated variable density groundwater flow with solute and energy transport', *US Geological Survey Water–Resources Investigations Report 02-4231*.

Wagner, BJ & Gorelick, SM 1987, 'Optimal groundwater quality management under parameter uncertainty', *Water Resources Research,* vol. 23**,** pp. 1162–1174.

Wagner, BJ & Gorelick, SM 1989, 'Reliable aquifer remediation in the presence of spatially-variable hydraulic conductivity – from data to design', *Water Resources Research,* vol. 25**,** pp. 2211–2225.

Wang, M & Zheng, C 1998, 'Ground water management optimization using genetic algorithms and simulated annealing: Formulation and comparison', *Journal of the American Water Resources Association,* vol. 34**,** pp. 519–530.

Wang, WC, Chau, KW, Cheng, CT & Qiu, L 2009, 'A comparison of performance of several artificial intelligence methods for forecasting monthly discharge time series', *Journal of Hydrology,* vol. 374**,** pp. 294–306.

Werner, AD 2010, 'A review of seawater intrusion and its management in Australia', *Hydrogeology Journal,* vol. 18**,** pp. 281–285.

Werner, AD & Gallagher, MR 2006, 'Characterisation of sea-water intrusion in the Pioneer Valley, Australia using hydrochemistry and three-dimensional numerical modelling', *Hydrogeology Journal,* vol. 14**,** pp. 1452–1469.

Werner, AD & Simmons, CT 2009, 'Impact of Sea-Level Rise on Sea Water Intrusion in Coastal Aquifers', *Ground Water,* vol. 47**,** pp. 197–204.

Willis, R & Finney, BA 1985, 'Optimal-control of nonlinear groundwater hydraulics – theoretical development and numerical experiments', *Water Resources Research,* vol. 21**,** pp. 1476–1482.

Wu, JF, Zheng, CM & Chien, CC 2005, 'Cost-effective sampling network design for contaminant plume monitoring under general hydrogeological conditions', *Journal of Contaminant Hydrology,* vol. 77**,** pp. 41–65.

Wu, JF, Zheng, CM, Chien, CC & Zheng, L. 2006, 'A comparative study of Monte Carlo simple genetic algorithm and noisy genetic algorithm for cost-effective sampling network design under uncertainty', *Advances in Water Resources,* vol. 29**,** pp. 899–911.

Yan, SQ & Minsker, B 2006, 'Optimal groundwater remediation design using an Adaptive Neural Network Genetic Algorithm', *Water Resources Research,* vol. 42**,** 14.

Yan, SQ & Minsker, B 2011a, 'Applying Dynamic Surrogate Models in Noisy Genetic Algorithms to Optimize Groundwater Remediation Designs', *Journal of Water Resources Planning and Management–ASCE,* vol. 137**,** pp. 284–292.

Zechman, E, Mirghani, B Mahinthakumar, G, & Ranjithan, SR 2005, 'A Genetic Programming-Based Surrogate Model Development and Its Application to a Groundwater Source Identification Problem', *ASCE Conference Proceedings*, 173, 341.

Zeitoun, DG & Dagan, G 1998, 'Benchmark solutions for seawater-freshwater interface problems in aquifers of random permeability distribution', *Computational Methods in Surface and Ground Water Transport: Proceedings of the 12th International Conference on Computational Methods in Water Resources*, vol. 2, 12**,** pp. 217–224.

Zhang, YQ, Pinder, GF & Herrera, GS 2005. Least cost design of groundwater quality monitoring networks. *Water Resources Research*, vol. 41.

Zhou, X, Chen, M & Liang, C 2003, 'Optimal schemes of groundwater exploitation for prevention of seawater intrusion in the Leizhou Peninsula in southern China', *Environmental Geology*, vol. 43**,** pp. 978–985.

Zhou, YX, Testroet, CBM & Vangeer, FC 1991, 'Using Kalman filtering to improve and quantify the uncertainty of numerical groundwater simulations 2. Application to monitoring network design', *Water Resources Research*, vol. 27**,** pp. 1995–2006.

Zouhri, L, Toto, E, Carlier, E & Debieche, TH 2010, 'Salinity of water resources: saltwater intrusion and water-rock interaction (western Morocco)', *Hydrological Sciences Journal–Journal Des Sciences Hydrologiques*, vol. 55**,** pp. 1337–1347.

APPENDICES

Appendix A: C code of GP model for salinity C1

```
double DiscipulusCFunction_1(double v[]);
#define TRUNC(x)(((x)>=0) ? floor(x) : ceil(x))
#define C_FPREM (_finite(f[0]/f[1]) ? f[0]-(TRUNC(f[0]/f[1])*f[1]) :
f[0]/f[1])
#define C_F2XM1 (((fabs(f[0])<=1) && (!_isnan(f[0]))) ? (pow(2,f[0])-
1) : ((!_finite(f[0]) && !_isnan(f[0]) && (f[0]<0)) ? -1 : f[0]))

double DiscipulusCFunction_1(double v[])
{
 long double f[8];
 long double tmp = 0;
 int cflag = 0;

 f[0]=f[1]=f[2]=f[3]=f[4]=f[5]=f[6]=f[7]=0;

 L0:  f[0]=cos(f[0]);
 L1:  f[0]/=0.1387641429901123f;
 L2:  cflag=(f[0] < f[2]);
 L3:  f[1]+=f[0];
 L4:  if (!cflag) f[0] = f[0];
 L5:  tmp=f[2]; f[2]=f[0]; f[0]=tmp;
 L6:  f[0]-=v[19];
 L7:  f[0]/=f[1];
 L8:  f[0]-=v[8];
 L9:  f[0]/=f[1];
 L10: f[0]+=f[0];
 L11: f[2]+=f[0];
 L12: f[0]/=f[0];
 L13: f[1]+=f[0];
 L14: f[0]=fabs(f[0]);
 L15: tmp=f[1]; f[1]=f[0]; f[0]=tmp;
 L16: f[0]=fabs(f[0]);
 L17: if (!cflag) f[0] = f[0];
 L18: f[0]*=f[0];
 L19: f[0]=-f[0];
 L20: f[2]-=f[0];
 L21: f[0]/=-0.09100413322448731f;
 L22: f[1]+=f[0];
 L23: f[2]+=f[0];
 L24: f[0]-=f[2];
 L25: f[0]+=f[1];
 L26: tmp=f[2]; f[2]=f[0]; f[0]=tmp;
 L27: f[0]+=f[1];
 L28: f[0]/=1.744837045669556f;
 L29: f[2]/=f[0];
 L30: f[0]/=1.744837045669556f;
 L31: tmp=f[1]; f[1]=f[0]; f[0]=tmp;
 L32: f[0]+=v[15];
 L33: f[0]=cos(f[0]);
 L34: f[1]+=f[0];
 L35: f[0]*=v[31];
 L36: f[0]+=v[11];
 L37: f[0]+=v[27];
```

```
 L38: f[0]+=v[27];
 L39: f[0]/=f[1];
 L40: f[0]+=f[2];
 L41: f[1]+=f[0];
 L42: f[0]=fabs(f[0]);
 L43: f[0]=sin(f[0]);
 L44: if (cflag) f[0] = f[0];
 L45: f[0]/=f[2];
 L46: f[1]+=f[0];
 L47: f[0]*=f[1];
 L48: f[0]-=v[22];
 L49: f[0]=fabs(f[0]);
 L50: tmp=f[0]; f[0]=f[0]; f[0]=tmp;
 L51: f[0]/=v[2];
 L52: f[0]+=v[5];
 L53: f[0]/=f[2];
 L54: cflag=(f[0] < f[1]);
 L55: cflag=(f[0] < f[2]);
 L56: f[0]+=v[16];
 L57: f[0]-=v[19];
 L58: f[0]+=f[0];
 L59: f[0]-=v[19];
 L60: f[0]+=f[0];
 L61: f[0]+=f[0];
 L62: f[0]+=f[2];
 L63: f[0]-=v[30];
 L64: f[0]*=1.248895406723023f;
 L65: f[0]+=f[0];
 L66: f[2]-=f[0];
 L67: f[0]/=f[1];
 L68: f[0]+=f[0];
 L69: f[0]+=f[1];
 L70:

 if (!_finite(f[0])) f[0]=0;
f[0]=f[0]+20;
 return f[0];
}

// Copyright, 2009, RML Technologies.
// This program was evolved with Discipulus(tm).
// This program and any information derived from this program
// may be used solely for pure research purposes and publication
// of results threrefrom in accordance with the Discipulus
// License agreement. This notice may not be removed from this
// program or any copy thereof.
```

Appendix B: C code of the MNN model to predict salinity C1

```
float Fire_Salinity_C1(float *inarray, float *outarray);
/* Insert this code into your C program to fire the C:\NeuroShell
2\EXAMPLES\Second models\Salinity C1\Salinity C1 network */
/* This code is designed to be simple and fast for porting to any
machine */
/* Therefore all code and weights are inline without looping or data
storage */
/*  which might be harder to port between compilers. */

#include <math.h>

float Fire_Salinity_C1(float *inarray, float *outarray)
{
 double netsum;
 double feature2[35];
 float c;
/* inarray[1] is C1 */
/* inarray[2] is C2 */
/* inarray[3] is C3 */
/* inarray[4] is C4 */
/* inarray[5] is C5 */
/* inarray[6] is C6 */
/* inarray[7] is C7 */
/* inarray[8] is C8 */
/* inarray[9] is C9 */
/* inarray[10] is C10 */
/* inarray[11] is C11 */
/* inarray[12] is C12 */
/* inarray[13] is C13 */
/* inarray[14] is C14 */
/* inarray[15] is C15 */
/* inarray[16] is C16 */
/* inarray[17] is C17 */
/* inarray[18] is C18 */
/* inarray[19] is C19 */
/* inarray[20] is C20 */
/* inarray[21] is C21 */
/* inarray[22] is C22 */
/* inarray[23] is C23 */
/* inarray[24] is C24 */
/* inarray[25] is C25 */
/* inarray[26] is C26 */
/* inarray[27] is C27 */
/* inarray[28] is C28 */
/* inarray[29] is C29 */
/* inarray[30] is C30 */
/* inarray[31] is C31 */
/* inarray[32] is C32 */
/* inarray[33] is C33 */
/* outarray[1] is C34 */

if (inarray[0]<0) inarray[0] = 0;
if (inarray[0]>1300) inarray[0] = 1300;
inarray[0] = inarray[0] / 1300;

if (inarray[1]<0) inarray[1] = 0;
if (inarray[1]>1300) inarray[1] = 1300;
```

```
inarray[1] = inarray[1] / 1300;

if (inarray[2]<0) inarray[2] = 0;
if (inarray[2]>1300) inarray[2] = 1300;
inarray[2] = inarray[2] / 1300;

if (inarray[3]<0) inarray[3] = 0;
if (inarray[3]>1300) inarray[3] = 1300;
inarray[3] = inarray[3] / 1300;

if (inarray[4]<0) inarray[4] = 0;
if (inarray[4]>1300) inarray[4] = 1300;
inarray[4] = inarray[4] / 1300;

if (inarray[5]<0) inarray[5] = 0;
if (inarray[5]>1300) inarray[5] = 1300;
inarray[5] = inarray[5] / 1300;

if (inarray[6]<0) inarray[6] = 0;
if (inarray[6]>1300) inarray[6] = 1300;
inarray[6] = inarray[6] / 1300;

if (inarray[7]<0) inarray[7] = 0;
if (inarray[7]>1300) inarray[7] = 1300;
inarray[7] = inarray[7] / 1300;

if (inarray[8]<0) inarray[8] = 0;
if (inarray[8]>1300) inarray[8] = 1300;
inarray[8] = inarray[8] / 1300;

if (inarray[9]<0) inarray[9] = 0;
if (inarray[9]>1300) inarray[9] = 1300;
inarray[9] = inarray[9] / 1300;

if (inarray[10]<0) inarray[10] = 0;
if (inarray[10]>1300) inarray[10] = 1300;
inarray[10] = inarray[10] / 1300;

if (inarray[11]<0) inarray[11] = 0;
if (inarray[11]>1300) inarray[11] = 1300;
inarray[11] = inarray[11] / 1300;

if (inarray[12]<0) inarray[12] = 0;
if (inarray[12]>1300) inarray[12] = 1300;
inarray[12] = inarray[12] / 1300;

if (inarray[13]<0) inarray[13] = 0;
if (inarray[13]>1300) inarray[13] = 1300;
inarray[13] = inarray[13] / 1300;

if (inarray[14]<0) inarray[14] = 0;
if (inarray[14]>1300) inarray[14] = 1300;
inarray[14] = inarray[14] / 1300;

if (inarray[15]<0) inarray[15] = 0;
if (inarray[15]>1300) inarray[15] = 1300;
inarray[15] = inarray[15] / 1300;

if (inarray[16]<0) inarray[16] = 0;
if (inarray[16]>1300) inarray[16] = 1300;
inarray[16] = inarray[16] / 1300;
```

```
if (inarray[17]<0) inarray[17] = 0;
if (inarray[17]>1300) inarray[17] = 1300;
inarray[17] = inarray[17] / 1300;

if (inarray[18]<0) inarray[18] = 0;
if (inarray[18]>1300) inarray[18] = 1300;
inarray[18] = inarray[18] / 1300;

if (inarray[19]<0) inarray[19] = 0;
if (inarray[19]>1300) inarray[19] = 1300;
inarray[19] = inarray[19] / 1300;

if (inarray[20]<0) inarray[20] = 0;
if (inarray[20]>1300) inarray[20] = 1300;
inarray[20] = inarray[20] / 1300;

if (inarray[21]<0) inarray[21] = 0;
if (inarray[21]>1300) inarray[21] = 1300;
inarray[21] = inarray[21] / 1300;

if (inarray[22]<0) inarray[22] = 0;
if (inarray[22]>1300) inarray[22] = 1300;
inarray[22] = inarray[22] / 1300;

if (inarray[23]<0) inarray[23] = 0;
if (inarray[23]>1300) inarray[23] = 1300;
inarray[23] = inarray[23] / 1300;

if (inarray[24]<0) inarray[24] = 0;
if (inarray[24]>1300) inarray[24] = 1300;
inarray[24] = inarray[24] / 1300;

if (inarray[25]<0) inarray[25] = 0;
if (inarray[25]>1300) inarray[25] = 1300;
inarray[25] = inarray[25] / 1300;

if (inarray[26]<0) inarray[26] = 0;
if (inarray[26]>1300) inarray[26] = 1300;
inarray[26] = inarray[26] / 1300;

if (inarray[27]<0) inarray[27] = 0;
if (inarray[27]>1300) inarray[27] = 1300;
inarray[27] = inarray[27] / 1300;

if (inarray[28]<0) inarray[28] = 0;
if (inarray[28]>1300) inarray[28] = 1300;
inarray[28] = inarray[28] / 1300;

if (inarray[29]<0) inarray[29] = 0;
if (inarray[29]>1300) inarray[29] = 1300;
inarray[29] = inarray[29] / 1300;

if (inarray[30]<0) inarray[30] = 0;
if (inarray[30]>1300) inarray[30] = 1300;
inarray[30] = inarray[30] / 1300;

if (inarray[31]<0) inarray[31] = 0;
if (inarray[31]>1300) inarray[31] = 1300;
inarray[31] = inarray[31] / 1300;
```

```
if (inarray[32]<0) inarray[32] = 0;
if (inarray[32]>1300) inarray[32] = 1300;
inarray[32] = inarray[32] / 1300;

netsum = -0.291164;
netsum += inarray[0] * 0.0609227;
netsum += inarray[1] * 0.1968207;
netsum += inarray[2] * -0.1315256;
netsum += inarray[3] * -3.741548E-02;
netsum += inarray[4] * 0.1272582;
netsum += inarray[5] * -0.2197175;
netsum += inarray[6] * -0.1866017;
netsum += inarray[7] * 2.244457E-02;
netsum += inarray[8] * -6.692994E-03;
netsum += inarray[9] * 0.219244;
netsum += inarray[10] * -0.2271028;
netsum += inarray[11] * 0.2581659;
netsum += inarray[12] * 0.1842414;
netsum += inarray[13] * 0.1516488;
netsum += inarray[14] * -6.242722E-02;
netsum += inarray[15] * 0.2175234;
netsum += inarray[16] * 9.198889E-02;
netsum += inarray[17] * 6.524261E-02;
netsum += inarray[18] * 0.1961007;
netsum += inarray[19] * -0.1924649;
netsum += inarray[20] * -6.918538E-02;
netsum += inarray[21] * -8.100132E-02;
netsum += inarray[22] * -0.1663927;
netsum += inarray[23] * 6.825688E-02;
netsum += inarray[24] * -0.1846358;
netsum += inarray[25] * -0.2380155;
netsum += inarray[26] * -0.1632848;
netsum += inarray[27] * -0.2373511;
netsum += inarray[28] * -0.1942585;
netsum += inarray[29] * -4.747362E-02;
netsum += inarray[30] * 0.2385039;
netsum += inarray[31] * 3.630452E-03;
netsum += inarray[32] * -0.1472658;
feature2[0] = 1 / (1 + exp(-netsum));

netsum = 1.174004E-02;
netsum += inarray[0] * 0.1062049;
netsum += inarray[1] * 0.1503448;
netsum += inarray[2] * 3.648394E-02;
netsum += inarray[3] * -8.159548E-04;
netsum += inarray[4] * 0.183258;
netsum += inarray[5] * -0.2927167;
netsum += inarray[6] * 3.191412E-02;
netsum += inarray[7] * -7.07048E-03;
netsum += inarray[8] * 0.1450065;
netsum += inarray[9] * 5.564062E-02;
netsum += inarray[10] * 0.1478936;
netsum += inarray[11] * 0.1796936;
netsum += inarray[12] * -8.575559E-02;
netsum += inarray[13] * 0.301464;
netsum += inarray[14] * -0.1997036;
netsum += inarray[15] * 8.759871E-02;
netsum += inarray[16] * -0.3099258;
netsum += inarray[17] * 0.1426232;
netsum += inarray[18] * 0.0613515;
netsum += inarray[19] * 0.5399987;
```

```
netsum += inarray[20] * -5.630424E-02;
netsum += inarray[21] * 0.2238498;
netsum += inarray[22] * -0.1239126;
netsum += inarray[23] * 9.933926E-02;
netsum += inarray[24] * -4.314826E-02;
netsum += inarray[25] * -2.091324E-02;
netsum += inarray[26] * 0.0433225;
netsum += inarray[27] * -0.2808856;
netsum += inarray[28] * -0.2073602;
netsum += inarray[29] * -0.2931437;
netsum += inarray[30] * 0.2953462;
netsum += inarray[31] * 4.099894E-02;
netsum += inarray[32] * 0.2630509;
feature2[1] = 1 / (1 + exp(-netsum));

netsum = -0.1730216;
netsum += inarray[0] * 0.1908191;
netsum += inarray[1] * 4.854375E-03;
netsum += inarray[2] * -0.2341436;
netsum += inarray[3] * 0.240606;
netsum += inarray[4] * 0.1497534;
netsum += inarray[5] * -0.1077782;
netsum += inarray[6] * 0.1354521;
netsum += inarray[7] * -4.134101E-02;
netsum += inarray[8] * 6.386779E-02;
netsum += inarray[9] * 0.1412297;
netsum += inarray[10] * -0.2121035;
netsum += inarray[11] * 4.108722E-02;
netsum += inarray[12] * 0.1298588;
netsum += inarray[13] * -0.1542069;
netsum += inarray[14] * -0.2622882;
netsum += inarray[15] * 0.1767584;
netsum += inarray[16] * 8.083119E-03;
netsum += inarray[17] * -5.671328E-02;
netsum += inarray[18] * -9.735201E-02;
netsum += inarray[19] * 0.0857999;
netsum += inarray[20] * -8.177768E-02;
netsum += inarray[21] * -4.610839E-02;
netsum += inarray[22] * 0.1678824;
netsum += inarray[23] * 8.731968E-02;
netsum += inarray[24] * -1.173465E-02;
netsum += inarray[25] * 6.770345E-02;
netsum += inarray[26] * -0.2292673;
netsum += inarray[27] * 0.1567391;
netsum += inarray[28] * -8.549603E-02;
netsum += inarray[29] * -0.1425688;
netsum += inarray[30] * -6.413723E-02;
netsum += inarray[31] * -0.1255498;
netsum += inarray[32] * 0.2090829;
feature2[2] = 1 / (1 + exp(-netsum));

netsum = 0.2024829;
netsum += inarray[0] * 1.394269E-02;
netsum += inarray[1] * -0.174963;
netsum += inarray[2] * 0.2696716;
netsum += inarray[3] * -0.130663;
netsum += inarray[4] * 0.0499515;
netsum += inarray[5] * 0.200725;
netsum += inarray[6] * 5.006719E-02;
netsum += inarray[7] * -0.1220727;
netsum += inarray[8] * 0.1631003;
```

```
netsum += inarray[9] * 0.2236539;
netsum += inarray[10] * -0.1251083;
netsum += inarray[11] * 0.2013144;
netsum += inarray[12] * -0.2191017;
netsum += inarray[13] * -2.801797E-02;
netsum += inarray[14] * 0.1679515;
netsum += inarray[15] * 0.2007551;
netsum += inarray[16] * -0.3317053;
netsum += inarray[17] * -8.138645E-02;
netsum += inarray[18] * -0.2834835;
netsum += inarray[19] * -4.429889E-02;
netsum += inarray[20] * -0.1972973;
netsum += inarray[21] * -0.176609;
netsum += inarray[22] * -7.453223E-02;
netsum += inarray[23] * 0.2208876;
netsum += inarray[24] * 0.2140077;
netsum += inarray[25] * 0.1034413;
netsum += inarray[26] * 0.2515329;
netsum += inarray[27] * -0.3207272;
netsum += inarray[28] * -1.507713E-02;
netsum += inarray[29] * -5.132416E-02;
netsum += inarray[30] * -0.1060462;
netsum += inarray[31] * 0.1896556;
netsum += inarray[32] * 0.3000493;
feature2[3] = 1 / (1 + exp(-netsum));

netsum = 6.836436E-02;
netsum += inarray[0] * -0.1245875;
netsum += inarray[1] * 0.1919254;
netsum += inarray[2] * 0.1506195;
netsum += inarray[3] * 9.57853E-03;
netsum += inarray[4] * 0.1760885;
netsum += inarray[5] * -0.2696576;
netsum += inarray[6] * -0.0836524;
netsum += inarray[7] * 1.64159E-03;
netsum += inarray[8] * 0.2342254;
netsum += inarray[9] * -0.2648467;
netsum += inarray[10] * 0.1916017;
netsum += inarray[11] * 0.1270443;
netsum += inarray[12] * 7.285731E-02;
netsum += inarray[13] * -0.2324152;
netsum += inarray[14] * -0.2502424;
netsum += inarray[15] * -0.1443405;
netsum += inarray[16] * -4.281345E-02;
netsum += inarray[17] * 0.2381991;
netsum += inarray[18] * -0.1627027;
netsum += inarray[19] * 0.3277583;
netsum += inarray[20] * 8.937006E-02;
netsum += inarray[21] * -0.2783402;
netsum += inarray[22] * 3.347053E-02;
netsum += inarray[23] * -0.1039962;
netsum += inarray[24] * -9.743807E-03;
netsum += inarray[25] * 9.249264E-02;
netsum += inarray[26] * 0.228207;
netsum += inarray[27] * -0.1232467;
netsum += inarray[28] * -0.1384581;
netsum += inarray[29] * -0.1728333;
netsum += inarray[30] * 0.1933142;
netsum += inarray[31] * 5.675236E-02;
netsum += inarray[32] * 0.1062352;
feature2[4] = 1 / (1 + exp(-netsum));
```

```
netsum = 6.80304E-03;
netsum += inarray[0] * 0.2264042;
netsum += inarray[1] * 2.654249E-03;
netsum += inarray[2] * -0.1951701;
netsum += inarray[3] * -0.132465;
netsum += inarray[4] * -0.1421895;
netsum += inarray[5] * 0.2636173;
netsum += inarray[6] * -1.962349E-03;
netsum += inarray[7] * 7.999559E-03;
netsum += inarray[8] * 0.1882886;
netsum += inarray[9] * -0.2183642;
netsum += inarray[10] * 0.2374518;
netsum += inarray[11] * -4.541584E-02;
netsum += inarray[12] * 0.1240751;
netsum += inarray[13] * 0.1630157;
netsum += inarray[14] * -0.2709817;
netsum += inarray[15] * -0.0725016;
netsum += inarray[16] * 0.1117489;
netsum += inarray[17] * 0.2731996;
netsum += inarray[18] * -0.1015783;
netsum += inarray[19] * 0.1510397;
netsum += inarray[20] * 0.1541723;
netsum += inarray[21] * 0.10844;
netsum += inarray[22] * 8.147637E-02;
netsum += inarray[23] * -1.296992E-02;
netsum += inarray[24] * -0.2893213;
netsum += inarray[25] * 0.2386213;
netsum += inarray[26] * 0.2105429;
netsum += inarray[27] * 0.1170489;
netsum += inarray[28] * 0.2583225;
netsum += inarray[29] * 0.2688383;
netsum += inarray[30] * 7.183626E-02;
netsum += inarray[31] * 5.496677E-02;
netsum += inarray[32] * 7.438053E-02;
feature2[5] = 1 / (1 + exp(-netsum));

netsum = -6.418992E-02;
netsum += inarray[0] * 0.1211364;
netsum += inarray[1] * 6.439745E-02;
netsum += inarray[2] * 0.284299;
netsum += inarray[3] * 0.3664744;
netsum += inarray[4] * -0.1680616;
netsum += inarray[5] * -0.417164;
netsum += inarray[6] * -0.2206384;
netsum += inarray[7] * -0.1771767;
netsum += inarray[8] * 0.2437314;
netsum += inarray[9] * 0.1636211;
netsum += inarray[10] * 0.2214312;
netsum += inarray[11] * -0.1086807;
netsum += inarray[12] * 0.1024779;
netsum += inarray[13] * -0.1384437;
netsum += inarray[14] * -0.157186;
netsum += inarray[15] * 0.1795875;
netsum += inarray[16] * -0.2552723;
netsum += inarray[17] * 0.1123518;
netsum += inarray[18] * -0.191859;
netsum += inarray[19] * 0.3384895;
netsum += inarray[20] * -5.115913E-02;
netsum += inarray[21] * 0.1760911;
netsum += inarray[22] * -0.2088975;
```

```
netsum += inarray[23] * -0.1288403;
netsum += inarray[24] * -0.106021;
netsum += inarray[25] * 5.286861E-02;
netsum += inarray[26] * -0.2857011;
netsum += inarray[27] * 0.213317;
netsum += inarray[28] * 0.1214973;
netsum += inarray[29] * 6.214068E-02;
netsum += inarray[30] * 0.1954981;
netsum += inarray[31] * -0.1284298;
netsum += inarray[32] * -0.2137673;
feature2[6] = 1 / (1 + exp(-netsum));

netsum = 0.2526204;
netsum += inarray[0] * 0.2550244;
netsum += inarray[1] * 0.3322617;
netsum += inarray[2] * 0.2200708;
netsum += inarray[3] * -8.354088E-02;
netsum += inarray[4] * -0.2747556;
netsum += inarray[5] * -2.578398E-02;
netsum += inarray[6] * 0.1043038;
netsum += inarray[7] * -9.431914E-02;
netsum += inarray[8] * 0.1438417;
netsum += inarray[9] * 6.840128E-02;
netsum += inarray[10] * 0.1474596;
netsum += inarray[11] * -0.2731026;
netsum += inarray[12] * -0.3058631;
netsum += inarray[13] * -3.146591E-02;
netsum += inarray[14] * -2.521872E-02;
netsum += inarray[15] * -0.1631775;
netsum += inarray[16] * 0.1051841;
netsum += inarray[17] * 0.3089578;
netsum += inarray[18] * 0.1445349;
netsum += inarray[19] * 0.3547677;
netsum += inarray[20] * 0.1130282;
netsum += inarray[21] * -0.2245904;
netsum += inarray[22] * 0.2925915;
netsum += inarray[23] * 0.2028049;
netsum += inarray[24] * -8.051314E-02;
netsum += inarray[25] * -0.1675046;
netsum += inarray[26] * -2.127141E-02;
netsum += inarray[27] * -8.755032E-03;
netsum += inarray[28] * 0.0511475;
netsum += inarray[29] * -0.1495467;
netsum += inarray[30] * 1.763342E-02;
netsum += inarray[31] * 0.1872907;
netsum += inarray[32] * -2.905235E-02;
feature2[7] = 1 / (1 + exp(-netsum));

netsum = 0.1796672;
netsum += inarray[0] * 0.3204595;
netsum += inarray[1] * 5.456817E-02;
netsum += inarray[2] * -1.255601E-02;
netsum += inarray[3] * 6.344802E-02;
netsum += inarray[4] * -0.3174484;
netsum += inarray[5] * -8.717819E-02;
netsum += inarray[6] * 3.819517E-02;
netsum += inarray[7] * 0.1492233;
netsum += inarray[8] * -0.298402;
netsum += inarray[9] * -0.2350557;
netsum += inarray[10] * 3.373973E-02;
netsum += inarray[11] * -6.876857E-02;
```

```
netsum += inarray[12] * 0.3293642;
netsum += inarray[13] * -0.2314587;
netsum += inarray[14] * -0.1089197;
netsum += inarray[15] * 3.650952E-02;
netsum += inarray[16] * 0.1463728;
netsum += inarray[17] * -0.298309;
netsum += inarray[18] * 1.921388E-02;
netsum += inarray[19] * -0.5553867;
netsum += inarray[20] * 0.2249069;
netsum += inarray[21] * -0.2711699;
netsum += inarray[22] * 0.1569463;
netsum += inarray[23] * 0.3539844;
netsum += inarray[24] * -0.2056857;
netsum += inarray[25] * 0.2750502;
netsum += inarray[26] * 0.1188893;
netsum += inarray[27] * 0.2569719;
netsum += inarray[28] * 0.1610216;
netsum += inarray[29] * 8.143257E-02;
netsum += inarray[30] * -0.4408813;
netsum += inarray[31] * -2.000959E-02;
netsum += inarray[32] * -0.1398602;
feature2[8] = 1 / (1 + exp(-netsum));

netsum = 0.1698739;
netsum += inarray[0] * 2.457304E-02;
netsum += inarray[1] * -0.2320768;
netsum += inarray[2] * -0.1882205;
netsum += inarray[3] * 4.164879E-02;
netsum += inarray[4] * 0.1886667;
netsum += inarray[5] * -9.60681E-03;
netsum += inarray[6] * 8.479623E-02;
netsum += inarray[7] * -0.1039934;
netsum += inarray[8] * -0.3062777;
netsum += inarray[9] * -0.2538242;
netsum += inarray[10] * -0.0687871;
netsum += inarray[11] * 0.2746648;
netsum += inarray[12] * 0.1085798;
netsum += inarray[13] * 0.2977886;
netsum += inarray[14] * -0.1743992;
netsum += inarray[15] * 4.365041E-02;
netsum += inarray[16] * 0.121898;
netsum += inarray[17] * 4.670427E-02;
netsum += inarray[18] * 0.2291728;
netsum += inarray[19] * 0.1103319;
netsum += inarray[20] * 2.431595E-02;
netsum += inarray[21] * -0.0610866;
netsum += inarray[22] * -0.20273;
netsum += inarray[23] * -0.1730192;
netsum += inarray[24] * 0.2016358;
netsum += inarray[25] * -0.0682168;
netsum += inarray[26] * 0.1531249;
netsum += inarray[27] * 0.0900219;
netsum += inarray[28] * 0.016502;
netsum += inarray[29] * 0.2310224;
netsum += inarray[30] * 0.231919;
netsum += inarray[31] * 0.1656791;
netsum += inarray[32] * 1.666201E-02;
feature2[9] = 1 / (1 + exp(-netsum));

netsum = -0.1774624;
netsum += inarray[0] * -0.1949999;
```

```
netsum += inarray[1] * 5.086683E-02;
netsum += inarray[2] * -0.19397;
netsum += inarray[3] * -0.1876334;
netsum += inarray[4] * 0.2092673;
netsum += inarray[5] * 0.3766194;
netsum += inarray[6] * 0.1302038;
netsum += inarray[7] * 0.2136044;
netsum += inarray[8] * -0.0104166;
netsum += inarray[9] * -0.2023746;
netsum += inarray[10] * 0.2086078;
netsum += inarray[11] * -6.118004E-02;
netsum += inarray[12] * 0.1564438;
netsum += inarray[13] * -0.2884678;
netsum += inarray[14] * -0.1552444;
netsum += inarray[15] * -2.586477E-02;
netsum += inarray[16] * 9.521113E-02;
netsum += inarray[17] * 0.2149922;
netsum += inarray[18] * -0.2043256;
netsum += inarray[19] * 0.115838;
netsum += inarray[20] * 1.680101E-02;
netsum += inarray[21] * 0.111361;
netsum += inarray[22] * -2.589812E-02;
netsum += inarray[23] * -7.788627E-02;
netsum += inarray[24] * 0.1357882;
netsum += inarray[25] * 0.3395701;
netsum += inarray[26] * 0.1835524;
netsum += inarray[27] * -0.1033115;
netsum += inarray[28] * 0.1393203;
netsum += inarray[29] * -0.1547541;
netsum += inarray[30] * -0.2742283;
netsum += inarray[31] * -7.160442E-02;
netsum += inarray[32] * -0.2692297;
feature2[10] = 1 / (1 + exp(-netsum));

netsum = 0.2217321;
netsum += inarray[0] * -2.245595E-02;
netsum += inarray[1] * -0.1246339;
netsum += inarray[2] * 0.1061224;
netsum += inarray[3] * 0.0972323;
netsum += inarray[4] * -0.2700748;
netsum += inarray[5] * -0.1363351;
netsum += inarray[6] * 0.2065502;
netsum += inarray[7] * 0.270154;
netsum += inarray[8] * -0.3924396;
netsum += inarray[9] * 2.990783E-02;
netsum += inarray[10] * -0.1175441;
netsum += inarray[11] * -0.1875806;
netsum += inarray[12] * -0.1946788;
netsum += inarray[13] * -0.2530636;
netsum += inarray[14] * 0.1483527;
netsum += inarray[15] * 0.1094594;
netsum += inarray[16] * -0.2091626;
netsum += inarray[17] * -3.269397E-02;
netsum += inarray[18] * 0.1871927;
netsum += inarray[19] * -0.6104613;
netsum += inarray[20] * -0.3334278;
netsum += inarray[21] * 0.1632083;
netsum += inarray[22] * 0.1644585;
netsum += inarray[23] * -8.894918E-02;
netsum += inarray[24] * 0.2210481;
netsum += inarray[25] * 0.1778918;
```

```
netsum += inarray[26] * -0.1034537;
netsum += inarray[27] * -5.660535E-02;
netsum += inarray[28] * 0.2452353;
netsum += inarray[29] * -0.343604;
netsum += inarray[30] * -0.2058159;
netsum += inarray[31] * -0.2497181;
netsum += inarray[32] * 0.223966;
feature2[11] = 1 / (1 + exp(-netsum));

netsum = 0.1488608;
netsum += inarray[0] * -0.1629222;
netsum += inarray[1] * 0.1720181;
netsum += inarray[2] * -1.033119E-02;
netsum += inarray[3] * 0.2143947;
netsum += inarray[4] * -0.2186943;
netsum += inarray[5] * 0.2124816;
netsum += inarray[6] * -0.2106929;
netsum += inarray[7] * 0.2484507;
netsum += inarray[8] * 0.1317291;
netsum += inarray[9] * -9.466735E-02;
netsum += inarray[10] * 0.2331553;
netsum += inarray[11] * 7.683255E-02;
netsum += inarray[12] * 3.301604E-02;
netsum += inarray[13] * -0.1619288;
netsum += inarray[14] * -4.910548E-02;
netsum += inarray[15] * 0.126996;
netsum += inarray[16] * 0.0190652;
netsum += inarray[17] * -8.688666E-02;
netsum += inarray[18] * 7.246707E-02;
netsum += inarray[19] * 0.1723192;
netsum += inarray[20] * -7.964721E-03;
netsum += inarray[21] * -0.1979022;
netsum += inarray[22] * 0.3247389;
netsum += inarray[23] * 0.1360425;
netsum += inarray[24] * 9.171806E-02;
netsum += inarray[25] * -0.1826937;
netsum += inarray[26] * 0.2449885;
netsum += inarray[27] * 0.2339713;
netsum += inarray[28] * 0.3031456;
netsum += inarray[29] * 0.1692172;
netsum += inarray[30] * -8.155958E-02;
netsum += inarray[31] * -1.437055E-02;
netsum += inarray[32] * -9.241096E-02;
feature2[12] = 1 / (1 + exp(-netsum));

netsum = -0.1726517;
netsum += inarray[0] * 0.4739076;
netsum += inarray[1] * -8.942349E-02;
netsum += inarray[2] * -0.4478568;
netsum += inarray[3] * -0.5703535;
netsum += inarray[4] * -0.4298367;
netsum += inarray[5] * -0.4740539;
netsum += inarray[6] * 3.582708E-02;
netsum += inarray[7] * -3.546358E-04;
netsum += inarray[8] * 0.8659835;
netsum += inarray[9] * 3.422061E-02;
netsum += inarray[10] * -0.3320189;
netsum += inarray[11] * -3.504724E-02;
netsum += inarray[12] * -0.1779616;
netsum += inarray[13] * -0.363067;
netsum += inarray[14] * 0.4969175;
```

```
netsum += inarray[15] * 0.5208058;
netsum += inarray[16] * -1.152801;
netsum += inarray[17] * -0.3191919;
netsum += inarray[18] * 0.7176239;
netsum += inarray[19] * 0.6429512;
netsum += inarray[20] * -0.1733667;
netsum += inarray[21] * -0.5916566;
netsum += inarray[22] * -5.932182E-02;
netsum += inarray[23] * 0.4216511;
netsum += inarray[24] * 0.2198181;
netsum += inarray[25] * 9.069564E-02;
netsum += inarray[26] * 0.3574776;
netsum += inarray[27] * -0.5911037;
netsum += inarray[28] * 0.2899445;
netsum += inarray[29] * 0.3835512;
netsum += inarray[30] * -0.2417199;
netsum += inarray[31] * 0.2460824;
netsum += inarray[32] * -0.3030266;
feature2[13] = 1 / (1 + exp(-netsum));

netsum = -0.1083389;
netsum += inarray[0] * -0.1667643;
netsum += inarray[1] * 4.164884E-02;
netsum += inarray[2] * -7.038092E-03;
netsum += inarray[3] * -0.0190748;
netsum += inarray[4] * -0.3036467;
netsum += inarray[5] * -0.190224;
netsum += inarray[6] * 0.2313906;
netsum += inarray[7] * -0.1228879;
netsum += inarray[8] * 5.182984E-02;
netsum += inarray[9] * -7.846867E-02;
netsum += inarray[10] * -0.1493768;
netsum += inarray[11] * 0.1064565;
netsum += inarray[12] * -0.25837;
netsum += inarray[13] * -0.2194721;
netsum += inarray[14] * 0.1081154;
netsum += inarray[15] * 0.1330149;
netsum += inarray[16] * 0.2634319;
netsum += inarray[17] * -0.2890002;
netsum += inarray[18] * -0.1837832;
netsum += inarray[19] * -0.152428;
netsum += inarray[20] * -1.718521E-02;
netsum += inarray[21] * -0.2642018;
netsum += inarray[22] * 9.232356E-02;
netsum += inarray[23] * 7.390402E-02;
netsum += inarray[24] * 2.774721E-02;
netsum += inarray[25] * -6.764226E-02;
netsum += inarray[26] * -0.1844685;
netsum += inarray[27] * -0.2599247;
netsum += inarray[28] * 0.2113642;
netsum += inarray[29] * 3.785059E-02;
netsum += inarray[30] * 6.488119E-02;
netsum += inarray[31] * -5.178367E-02;
netsum += inarray[32] * 0.2692251;
feature2[14] = 1 / (1 + exp(-netsum));

netsum = -0.3776446;
netsum += inarray[0] * 0.1597989;
netsum += inarray[1] * -0.2023886;
netsum += inarray[2] * 1.740474E-02;
netsum += inarray[3] * 0.1112997;
```

```
netsum += inarray[4] * -0.3100364;
netsum += inarray[5] * 0.3348331;
netsum += inarray[6] * 0.2652803;
netsum += inarray[7] * 0.1528982;
netsum += inarray[8] * -0.1707719;
netsum += inarray[9] * 7.758281E-02;
netsum += inarray[10] * -3.991274E-02;
netsum += inarray[11] * 0.2652445;
netsum += inarray[12] * -6.619821E-02;
netsum += inarray[13] * -0.1433327;
netsum += inarray[14] * -1.600052E-02;
netsum += inarray[15] * -0.1745892;
netsum += inarray[16] * 8.126199E-02;
netsum += inarray[17] * -3.955205E-02;
netsum += inarray[18] * -1.189258E-02;
netsum += inarray[19] * -0.96208;
netsum += inarray[20] * 0.1291979;
netsum += inarray[21] * -0.1747367;
netsum += inarray[22] * 2.067315E-02;
netsum += inarray[23] * -0.1145532;
netsum += inarray[24] * 6.579524E-02;
netsum += inarray[25] * -0.1941968;
netsum += inarray[26] * 1.415176E-02;
netsum += inarray[27] * -0.207614;
netsum += inarray[28] * -0.2224153;
netsum += inarray[29] * 0.1180836;
netsum += inarray[30] * 0.1728862;
netsum += inarray[31] * -5.487831E-02;
netsum += inarray[32] * 7.854225E-04;
feature2[15] = 1 / (1 + exp(-netsum));

netsum = -0.2313401;
netsum += inarray[0] * 0.2126334;
netsum += inarray[1] * -0.2064492;
netsum += inarray[2] * -6.115143E-02;
netsum += inarray[3] * -0.2902556;
netsum += inarray[4] * -1.653757E-02;
netsum += inarray[5] * -2.684269E-02;
netsum += inarray[6] * 7.921445E-02;
netsum += inarray[7] * -0.2279978;
netsum += inarray[8] * -1.993167E-02;
netsum += inarray[9] * 0.1873854;
netsum += inarray[10] * 0.303654;
netsum += inarray[11] * -3.559806E-02;
netsum += inarray[12] * -0.1922244;
netsum += inarray[13] * 0.2161546;
netsum += inarray[14] * 0.1245562;
netsum += inarray[15] * -0.1819935;
netsum += inarray[16] * 0.3110418;
netsum += inarray[17] * -0.0726958;
netsum += inarray[18] * 0.2556754;
netsum += inarray[19] * -0.1839936;
netsum += inarray[20] * -0.0101628;
netsum += inarray[21] * -0.2814794;
netsum += inarray[22] * 0.2711261;
netsum += inarray[23] * 0.1064773;
netsum += inarray[24] * 0.1572417;
netsum += inarray[25] * 5.262619E-02;
netsum += inarray[26] * -0.2729664;
netsum += inarray[27] * -0.2073185;
netsum += inarray[28] * -7.298999E-02;
```

```
netsum += inarray[29] * 0.2079828;
netsum += inarray[30] * 6.403983E-02;
netsum += inarray[31] * -0.2707055;
netsum += inarray[32] * -0.1595489;
feature2[16] = 1 / (1 + exp(-netsum));

netsum = 3.661797E-02;
netsum += inarray[0] * 0.2185052;
netsum += inarray[1] * -1.313948E-02;
netsum += inarray[2] * -7.461892E-02;
netsum += inarray[3] * 0.4108629;
netsum += inarray[4] * 0.2086206;
netsum += inarray[5] * 8.901048E-02;
netsum += inarray[6] * 0.1169733;
netsum += inarray[7] * -0.1715256;
netsum += inarray[8] * -0.4502749;
netsum += inarray[9] * 0.1530477;
netsum += inarray[10] * -0.2031143;
netsum += inarray[11] * 0.2065634;
netsum += inarray[12] * 0.4939364;
netsum += inarray[13] * 0.3767501;
netsum += inarray[14] * -0.1656672;
netsum += inarray[15] * -0.1676554;
netsum += inarray[16] * 0.497452;
netsum += inarray[17] * -0.2319279;
netsum += inarray[18] * 5.456683E-02;
netsum += inarray[19] * -0.1879045;
netsum += inarray[20] * -0.1923237;
netsum += inarray[21] * 0.3947749;
netsum += inarray[22] * 0.1737501;
netsum += inarray[23] * -0.1498373;
netsum += inarray[24] * 0.1921258;
netsum += inarray[25] * -0.2326475;
netsum += inarray[26] * 3.836017E-02;
netsum += inarray[27] * 0.4546024;
netsum += inarray[28] * -5.827418E-02;
netsum += inarray[29] * -0.1738803;
netsum += inarray[30] * 0.2292359;
netsum += inarray[31] * -0.1288378;
netsum += inarray[32] * 1.373574E-02;
feature2[17] = 1 / (1 + exp(-netsum));

netsum = -0.1789295;
netsum += inarray[0] * 0.3271207;
netsum += inarray[1] * -3.543137E-02;
netsum += inarray[2] * 6.573328E-02;
netsum += inarray[3] * 5.618945E-02;
netsum += inarray[4] * 0.1324985;
netsum += inarray[5] * 0.1453875;
netsum += inarray[6] * -0.2473926;
netsum += inarray[7] * 0.2064603;
netsum += inarray[8] * 0.1045483;
netsum += inarray[9] * 3.317058E-02;
netsum += inarray[10] * -0.2437914;
netsum += inarray[11] * -0.2697509;
netsum += inarray[12] * 9.710424E-02;
netsum += inarray[13] * -0.1638602;
netsum += inarray[14] * 0.2385335;
netsum += inarray[15] * 0.1751873;
netsum += inarray[16] * 7.164589E-02;
netsum += inarray[17] * -0.2796746;
```

```
netsum += inarray[18] * 0.3185912;
netsum += inarray[19] * 0.1126974;
netsum += inarray[20] * 2.101329E-02;
netsum += inarray[21] * 9.171607E-02;
netsum += inarray[22] * -0.123794;
netsum += inarray[23] * 0.0794915;
netsum += inarray[24] * -2.478753E-02;
netsum += inarray[25] * 0.308364;
netsum += inarray[26] * 0.2044287;
netsum += inarray[27] * -7.963759E-02;
netsum += inarray[28] * -0.1854234;
netsum += inarray[29] * 6.169935E-02;
netsum += inarray[30] * -5.585483E-02;
netsum += inarray[31] * 7.533608E-02;
netsum += inarray[32] * 3.327246E-02;
feature2[18] = 1 / (1 + exp(-netsum));

netsum = -0.2819329;
netsum += inarray[0] * 3.099443E-02;
netsum += inarray[1] * 0.0502328;
netsum += inarray[2] * 2.573386E-02;
netsum += inarray[3] * -0.3390821;
netsum += inarray[4] * -2.609201E-02;
netsum += inarray[5] * 0.1998909;
netsum += inarray[6] * 6.478088E-02;
netsum += inarray[7] * -4.023553E-02;
netsum += inarray[8] * -0.2331954;
netsum += inarray[9] * 3.536383E-03;
netsum += inarray[10] * -4.175213E-02;
netsum += inarray[11] * 0.163009;
netsum += inarray[12] * -8.526818E-02;
netsum += inarray[13] * 4.985042E-02;
netsum += inarray[14] * 0.1187532;
netsum += inarray[15] * 0.2915817;
netsum += inarray[16] * 0.1939631;
netsum += inarray[17] * -6.729486E-02;
netsum += inarray[18] * -0.181965;
netsum += inarray[19] * -0.3403972;
netsum += inarray[20] * 5.060034E-02;
netsum += inarray[21] * -0.3140913;
netsum += inarray[22] * -2.067569E-02;
netsum += inarray[23] * 0.0626684;
netsum += inarray[24] * -9.946167E-02;
netsum += inarray[25] * -0.1578715;
netsum += inarray[26] * 0.1056338;
netsum += inarray[27] * -0.1293214;
netsum += inarray[28] * 0.1869922;
netsum += inarray[29] * 0.188171;
netsum += inarray[30] * 6.571969E-02;
netsum += inarray[31] * 0.1984921;
netsum += inarray[32] * 0.1666871;
feature2[19] = 1 / (1 + exp(-netsum));

netsum = 0.3177998;
netsum += inarray[0] * -6.480873E-02;
netsum += inarray[1] * -5.863044E-03;
netsum += inarray[2] * 0.1604046;
netsum += inarray[3] * 0.0489388;
netsum += inarray[4] * -0.2111368;
netsum += inarray[5] * 9.895658E-02;
netsum += inarray[6] * 0.2742001;
```

```
netsum += inarray[7] * 8.426016E-02;
netsum += inarray[8] * -0.2715441;
netsum += inarray[9] * 0.1421343;
netsum += inarray[10] * -3.240113E-02;
netsum += inarray[11] * 2.320839E-03;
netsum += inarray[12] * 0.3614705;
netsum += inarray[13] * 0.1183433;
netsum += inarray[14] * -0.2007138;
netsum += inarray[15] * -7.695301E-02;
netsum += inarray[16] * 0.2788737;
netsum += inarray[17] * 0.2125321;
netsum += inarray[18] * 0.2286511;
netsum += inarray[19] * -0.2572988;
netsum += inarray[20] * 0.1245328;
netsum += inarray[21] * 0.3076485;
netsum += inarray[22] * -0.1046421;
netsum += inarray[23] * -1.047813E-02;
netsum += inarray[24] * -0.2160057;
netsum += inarray[25] * -4.699736E-02;
netsum += inarray[26] * -0.2804104;
netsum += inarray[27] * 0.2409735;
netsum += inarray[28] * 9.923988E-02;
netsum += inarray[29] * 0.1206731;
netsum += inarray[30] * 0.1048922;
netsum += inarray[31] * -6.486004E-02;
netsum += inarray[32] * -0.1089544;
feature2[20] = 1 / (1 + exp(-netsum));

netsum = -2.885145E-02;
netsum += inarray[0] * -0.1452485;
netsum += inarray[1] * 0.3607551;
netsum += inarray[2] * 0.1854913;
netsum += inarray[3] * -0.1149906;
netsum += inarray[4] * 3.786818E-02;
netsum += inarray[5] * -3.821171E-02;
netsum += inarray[6] * -0.1245146;
netsum += inarray[7] * 0.3082392;
netsum += inarray[8] * 0.2471695;
netsum += inarray[9] * 0.2420119;
netsum += inarray[10] * -0.255326;
netsum += inarray[11] * 0.2153737;
netsum += inarray[12] * -7.234699E-02;
netsum += inarray[13] * 0.1967219;
netsum += inarray[14] * 0.2051766;
netsum += inarray[15] * 2.680431E-02;
netsum += inarray[16] * -3.862504E-02;
netsum += inarray[17] * 0.2940324;
netsum += inarray[18] * 0.125513;
netsum += inarray[19] * 0.2251876;
netsum += inarray[20] * 0.1979208;
netsum += inarray[21] * 0.1239398;
netsum += inarray[22] * 2.501711E-02;
netsum += inarray[23] * -0.3060785;
netsum += inarray[24] * 0.2257033;
netsum += inarray[25] * -0.1456783;
netsum += inarray[26] * 0.1439563;
netsum += inarray[27] * -0.1955269;
netsum += inarray[28] * 0.1555102;
netsum += inarray[29] * 9.991781E-02;
netsum += inarray[30] * -0.2355822;
netsum += inarray[31] * 0.2229307;
```

```
netsum += inarray[32] * -0.1122014;
feature2[21] = 1 / (1 + exp(-netsum));

netsum = -0.24348;
netsum += inarray[0] * -0.135895;
netsum += inarray[1] * 0.1705584;
netsum += inarray[2] * -5.277328E-03;
netsum += inarray[3] * 0.2079019;
netsum += inarray[4] * -0.1704331;
netsum += inarray[5] * -9.330103E-02;
netsum += inarray[6] * 2.338426E-02;
netsum += inarray[7] * -0.2693043;
netsum += inarray[8] * 0.2308157;
netsum += inarray[9] * -0.2068782;
netsum += inarray[10] * -0.1534795;
netsum += inarray[11] * 4.381708E-02;
netsum += inarray[12] * 8.618804E-02;
netsum += inarray[13] * 0.2622083;
netsum += inarray[14] * 0.1773846;
netsum += inarray[15] * -0.2972995;
netsum += inarray[16] * -0.1616764;
netsum += inarray[17] * -6.050858E-02;
netsum += inarray[18] * 0.2471552;
netsum += inarray[19] * 0.0616068;
netsum += inarray[20] * 0.2398393;
netsum += inarray[21] * -2.876074E-02;
netsum += inarray[22] * 0.1433202;
netsum += inarray[23] * 0.2310338;
netsum += inarray[24] * 0.2040362;
netsum += inarray[25] * -0.2620143;
netsum += inarray[26] * 1.868654E-02;
netsum += inarray[27] * 0.1667046;
netsum += inarray[28] * -0.2719508;
netsum += inarray[29] * -0.1535261;
netsum += inarray[30] * 6.819336E-02;
netsum += inarray[31] * -0.1061908;
netsum += inarray[32] * 0.1506103;
feature2[22] = 1 / (1 + exp(-netsum));

netsum = -0.1725512;
netsum += inarray[0] * 0.2751062;
netsum += inarray[1] * -0.130754;
netsum += inarray[2] * -0.2940531;
netsum += inarray[3] * 1.403669E-02;
netsum += inarray[4] * 8.086415E-02;
netsum += inarray[5] * 0.4970524;
netsum += inarray[6] * -0.0443826;
netsum += inarray[7] * 0.3333143;
netsum += inarray[8] * -0.5215071;
netsum += inarray[9] * 2.080907E-02;
netsum += inarray[10] * 0.2957908;
netsum += inarray[11] * -0.1697735;
netsum += inarray[12] * -6.102575E-02;
netsum += inarray[13] * -0.0959894;
netsum += inarray[14] * 0.1353661;
netsum += inarray[15] * 0.3631852;
netsum += inarray[16] * -6.503319E-02;
netsum += inarray[17] * 0.1776098;
netsum += inarray[18] * 0.2794216;
netsum += inarray[19] * -0.4548127;
netsum += inarray[20] * -0.1553866;
```

```
netsum += inarray[21] * -0.2472495;
netsum += inarray[22] * -9.655437E-02;
netsum += inarray[23] * 2.739238E-02;
netsum += inarray[24] * 3.372616E-02;
netsum += inarray[25] * 8.809957E-03;
netsum += inarray[26] * 0.3261434;
netsum += inarray[27] * -2.402057E-02;
netsum += inarray[28] * 0.1082372;
netsum += inarray[29] * 0.1954163;
netsum += inarray[30] * -0.4688437;
netsum += inarray[31] * 9.525979E-02;
netsum += inarray[32] * 0.2550399;
feature2[23] = 1 / (1 + exp(-netsum));

netsum = -0.2747058;
netsum += inarray[0] * -0.305579;
netsum += inarray[1] * -0.21069;
netsum += inarray[2] * -0.1577966;
netsum += inarray[3] * 1.415811E-02;
netsum += inarray[4] * -0.1888796;
netsum += inarray[5] * 0.2907608;
netsum += inarray[6] * -0.2319607;
netsum += inarray[7] * 0.1300429;
netsum += inarray[8] * -0.1114993;
netsum += inarray[9] * -9.227605E-02;
netsum += inarray[10] * -0.1211003;
netsum += inarray[11] * -0.1740388;
netsum += inarray[12] * -0.1903145;
netsum += inarray[13] * 0.2104421;
netsum += inarray[14] * 0.3137967;
netsum += inarray[15] * -0.2241699;
netsum += inarray[16] * 0.1488807;
netsum += inarray[17] * 0.1218716;
netsum += inarray[18] * 0.2668924;
netsum += inarray[19] * -0.2348025;
netsum += inarray[20] * 0.2027763;
netsum += inarray[21] * -0.132873;
netsum += inarray[22] * 4.033266E-02;
netsum += inarray[23] * 0.1263477;
netsum += inarray[24] * -0.2365156;
netsum += inarray[25] * -0.2629387;
netsum += inarray[26] * 0.2181934;
netsum += inarray[27] * 0.2107381;
netsum += inarray[28] * 0.1876403;
netsum += inarray[29] * -0.1195808;
netsum += inarray[30] * -0.3373014;
netsum += inarray[31] * -0.2693075;
netsum += inarray[32] * -0.1428874;
feature2[24] = 1 / (1 + exp(-netsum));

netsum = -2.268392E-02;
netsum += inarray[0] * 7.004675E-03;
netsum += inarray[1] * 6.873157E-02;
netsum += inarray[2] * 0.1111907;
netsum += inarray[3] * -0.2524659;
netsum += inarray[4] * -9.460418E-02;
netsum += inarray[5] * 0.5971823;
netsum += inarray[6] * -0.2295774;
netsum += inarray[7] * -0.2020459;
netsum += inarray[8] * -0.5944926;
netsum += inarray[9] * -0.1497179;
```

```
netsum += inarray[10] * -0.2080041;
netsum += inarray[11] * -4.301437E-03;
netsum += inarray[12] * 8.981773E-03;
netsum += inarray[13] * -0.1052709;
netsum += inarray[14] * 0.0587485;
netsum += inarray[15] * 7.898964E-02;
netsum += inarray[16] * 6.167773E-02;
netsum += inarray[17] * 0.2086655;
netsum += inarray[18] * 7.707877E-02;
netsum += inarray[19] * -0.8609802;
netsum += inarray[20] * 0.1187937;
netsum += inarray[21] * -6.510927E-02;
netsum += inarray[22] * -0.0101573;
netsum += inarray[23] * -0.033094;
netsum += inarray[24] * 0.2418814;
netsum += inarray[25] * 0.289637;
netsum += inarray[26] * 0.1993061;
netsum += inarray[27] * -0.1019313;
netsum += inarray[28] * 0.1087175;
netsum += inarray[29] * -0.1502966;
netsum += inarray[30] * -1.625599E-02;
netsum += inarray[31] * 0.4358023;
netsum += inarray[32] * -0.2807171;
feature2[25] = 1 / (1 + exp(-netsum));

netsum = 0.1385553;
netsum += inarray[0] * -0.1824679;
netsum += inarray[1] * 7.111401E-02;
netsum += inarray[2] * -0.125566;
netsum += inarray[3] * 0.4543407;
netsum += inarray[4] * -2.229035E-02;
netsum += inarray[5] * -7.340672E-02;
netsum += inarray[6] * 0.1840906;
netsum += inarray[7] * -0.117557;
netsum += inarray[8] * 0.3456187;
netsum += inarray[9] * -0.191919;
netsum += inarray[10] * -0.3492042;
netsum += inarray[11] * -0.3516404;
netsum += inarray[12] * 0.0985539;
netsum += inarray[13] * 5.101077E-02;
netsum += inarray[14] * 0.1174238;
netsum += inarray[15] * 0.2264117;
netsum += inarray[16] * -0.3396508;
netsum += inarray[17] * 0.3400194;
netsum += inarray[18] * -0.2028574;
netsum += inarray[19] * 0.9176918;
netsum += inarray[20] * 0.2428584;
netsum += inarray[21] * 0.2610558;
netsum += inarray[22] * 0.394118;
netsum += inarray[23] * -0.3563049;
netsum += inarray[24] * 0.1683055;
netsum += inarray[25] * 0.209172;
netsum += inarray[26] * 4.950962E-03;
netsum += inarray[27] * -0.1445089;
netsum += inarray[28] * 0.3038435;
netsum += inarray[29] * -0.1415799;
netsum += inarray[30] * 6.109403E-02;
netsum += inarray[31] * 0.0754426;
netsum += inarray[32] * 0.1134597;
feature2[26] = 1 / (1 + exp(-netsum));
```

```
netsum = 0.1183951;
netsum += inarray[0] * -1.874938E-02;
netsum += inarray[1] * -0.3596035;
netsum += inarray[2] * 0.2128175;
netsum += inarray[3] * -3.986773E-02;
netsum += inarray[4] * 0.1570608;
netsum += inarray[5] * -0.2754308;
netsum += inarray[6] * 0.3172037;
netsum += inarray[7] * 9.670895E-02;
netsum += inarray[8] * 0.740764;
netsum += inarray[9] * 0.2245852;
netsum += inarray[10] * 0.1225005;
netsum += inarray[11] * -0.1417626;
netsum += inarray[12] * 0.1936337;
netsum += inarray[13] * 0.259302;
netsum += inarray[14] * -0.1912074;
netsum += inarray[15] * -5.332145E-03;
netsum += inarray[16] * -0.3689941;
netsum += inarray[17] * -8.094183E-03;
netsum += inarray[18] * 3.657931E-02;
netsum += inarray[19] * 0.6399341;
netsum += inarray[20] * 0.2676957;
netsum += inarray[21] * 0.3025043;
netsum += inarray[22] * 0.2253492;
netsum += inarray[23] * -0.1895468;
netsum += inarray[24] * -0.2825996;
netsum += inarray[25] * 0.043952;
netsum += inarray[26] * -0.027178;
netsum += inarray[27] * -0.2522345;
netsum += inarray[28] * -0.2087769;
netsum += inarray[29] * 0.1007466;
netsum += inarray[30] * 0.5667402;
netsum += inarray[31] * -0.289344;
netsum += inarray[32] * 0.4443603;
feature2[27] = 1 / (1 + exp(-netsum));

netsum = 6.554653E-02;
netsum += inarray[0] * -0.1964628;
netsum += inarray[1] * 0.1263493;
netsum += inarray[2] * -3.009215E-02;
netsum += inarray[3] * -0.3823033;
netsum += inarray[4] * -6.088189E-02;
netsum += inarray[5] * 0.2023286;
netsum += inarray[6] * 0.1741232;
netsum += inarray[7] * -0.2255233;
netsum += inarray[8] * -0.0767581;
netsum += inarray[9] * -0.2984591;
netsum += inarray[10] * -0.2098258;
netsum += inarray[11] * -0.1972663;
netsum += inarray[12] * -0.1279699;
netsum += inarray[13] * -0.1088066;
netsum += inarray[14] * 3.208917E-02;
netsum += inarray[15] * 0.1497795;
netsum += inarray[16] * 0.2517925;
netsum += inarray[17] * 1.592465E-02;
netsum += inarray[18] * -0.1438153;
netsum += inarray[19] * -0.751308;
netsum += inarray[20] * 0.277462;
netsum += inarray[21] * -0.3061544;
netsum += inarray[22] * 0.1377325;
netsum += inarray[23] * 0.046709;
```

```
netsum += inarray[24] * -2.737018E-02;
netsum += inarray[25] * -0.263502;
netsum += inarray[26] * -2.200815E-02;
netsum += inarray[27] * -0.1386736;
netsum += inarray[28] * 0.1395172;
netsum += inarray[29] * -3.311163E-02;
netsum += inarray[30] * -7.306328E-02;
netsum += inarray[31] * 0.2131454;
netsum += inarray[32] * -5.823568E-04;
feature2[28] = 1 / (1 + exp(-netsum));

netsum = 1.611936E-02;
netsum += inarray[0] * -0.3796925;
netsum += inarray[1] * 9.447172E-02;
netsum += inarray[2] * -5.387916E-02;
netsum += inarray[3] * 0.2073087;
netsum += inarray[4] * 0.103752;
netsum += inarray[5] * -0.2383531;
netsum += inarray[6] * -0.2428635;
netsum += inarray[7] * -0.1753693;
netsum += inarray[8] * -0.1674487;
netsum += inarray[9] * -4.169188E-02;
netsum += inarray[10] * -0.2501166;
netsum += inarray[11] * 0.1800583;
netsum += inarray[12] * 0.1789062;
netsum += inarray[13] * -3.576826E-02;
netsum += inarray[14] * -6.690817E-02;
netsum += inarray[15] * -0.3810511;
netsum += inarray[16] * -0.2318026;
netsum += inarray[17] * -0.2373058;
netsum += inarray[18] * -0.2831336;
netsum += inarray[19] * 0.3430523;
netsum += inarray[20] * 0.240981;
netsum += inarray[21] * 0.2356786;
netsum += inarray[22] * 0.1092813;
netsum += inarray[23] * -0.2007592;
netsum += inarray[24] * 0.2389184;
netsum += inarray[25] * -0.1789904;
netsum += inarray[26] * -0.3327489;
netsum += inarray[27] * 0.2821064;
netsum += inarray[28] * -0.1000369;
netsum += inarray[29] * -0.2341267;
netsum += inarray[30] * 0.21124;
netsum += inarray[31] * -0.1345972;
netsum += inarray[32] * -0.2516265;
feature2[29] = 1 / (1 + exp(-netsum));

netsum = -0.1840916;
netsum += inarray[0] * 0.268027;
netsum += inarray[1] * -0.1840145;
netsum += inarray[2] * -0.2186075;
netsum += inarray[3] * -9.790255E-02;
netsum += inarray[4] * -0.1797517;
netsum += inarray[5] * -0.3578763;
netsum += inarray[6] * -0.0878347;
netsum += inarray[7] * 0.2608691;
netsum += inarray[8] * 0.111102;
netsum += inarray[9] * 7.992725E-02;
netsum += inarray[10] * -0.2570114;
netsum += inarray[11] * 0.2250037;
netsum += inarray[12] * -5.926724E-02;
```

```
netsum += inarray[13] * 0.29929;
netsum += inarray[14] * 9.792642E-02;
netsum += inarray[15] * -0.2853814;
netsum += inarray[16] * -0.1929283;
netsum += inarray[17] * 0.186277;
netsum += inarray[18] * 0.1343688;
netsum += inarray[19] * 7.536424E-02;
netsum += inarray[20] * 0.3168816;
netsum += inarray[21] * -0.2115398;
netsum += inarray[22] * -1.030523E-03;
netsum += inarray[23] * 2.594086E-02;
netsum += inarray[24] * -0.2178956;
netsum += inarray[25] * -4.666506E-02;
netsum += inarray[26] * 9.516257E-02;
netsum += inarray[27] * 0.2077898;
netsum += inarray[28] * 0.1172324;
netsum += inarray[29] * -7.985284E-02;
netsum += inarray[30] * -0.1999155;
netsum += inarray[31] * 4.760027E-02;
netsum += inarray[32] * 0.3108743;
feature2[30] = 1 / (1 + exp(-netsum));

netsum = 0.1095111;
netsum += inarray[0] * -0.1424592;
netsum += inarray[1] * 0.2515203;
netsum += inarray[2] * 0.2428484;
netsum += inarray[3] * 0.2330756;
netsum += inarray[4] * -1.485163E-02;
netsum += inarray[5] * 0.1224327;
netsum += inarray[6] * -0.255242;
netsum += inarray[7] * 2.242427E-02;
netsum += inarray[8] * -0.2022949;
netsum += inarray[9] * -0.1246995;
netsum += inarray[10] * -8.931931E-02;
netsum += inarray[11] * -0.2810515;
netsum += inarray[12] * -0.2406703;
netsum += inarray[13] * 2.637865E-03;
netsum += inarray[14] * -0.2705601;
netsum += inarray[15] * -0.281466;
netsum += inarray[16] * -2.162175E-02;
netsum += inarray[17] * -2.286173E-02;
netsum += inarray[18] * 0.1838146;
netsum += inarray[19] * -0.2308593;
netsum += inarray[20] * 0.2752168;
netsum += inarray[21] * 8.664005E-02;
netsum += inarray[22] * 0.1785185;
netsum += inarray[23] * -5.542005E-02;
netsum += inarray[24] * -0.1820633;
netsum += inarray[25] * 0.1731886;
netsum += inarray[26] * -0.3137177;
netsum += inarray[27] * 0.192462;
netsum += inarray[28] * 0.2485198;
netsum += inarray[29] * 0.2048407;
netsum += inarray[30] * 0.1116911;
netsum += inarray[31] * 5.757286E-03;
netsum += inarray[32] * -0.2417923;
feature2[31] = 1 / (1 + exp(-netsum));

netsum = -1.216965E-02;
netsum += inarray[0] * -0.1492647;
netsum += inarray[1] * -0.2022796;
```

```
netsum += inarray[2] * 0.2932994;
netsum += inarray[3] * -0.1081511;
netsum += inarray[4] * 0.3212546;
netsum += inarray[5] * -0.1964692;
netsum += inarray[6] * -0.2387215;
netsum += inarray[7] * 0.1388554;
netsum += inarray[8] * 0.3854406;
netsum += inarray[9] * 7.352878E-02;
netsum += inarray[10] * 0.2647846;
netsum += inarray[11] * 0.2037821;
netsum += inarray[12] * 0.1263289;
netsum += inarray[13] * 0.1068067;
netsum += inarray[14] * -0.1996129;
netsum += inarray[15] * -0.3406896;
netsum += inarray[16] * -0.4481568;
netsum += inarray[17] * 8.951739E-02;
netsum += inarray[18] * -0.2008466;
netsum += inarray[19] * 9.022786E-02;
netsum += inarray[20] * 9.792662E-02;
netsum += inarray[21] * 5.548611E-02;
netsum += inarray[22] * 0.2058237;
netsum += inarray[23] * -0.1381858;
netsum += inarray[24] * -0.1594197;
netsum += inarray[25] * -8.377776E-02;
netsum += inarray[26] * 0.1780602;
netsum += inarray[27] * 0.188692;
netsum += inarray[28] * 0.2695051;
netsum += inarray[29] * -0.1618377;
netsum += inarray[30] * -0.1655446;
netsum += inarray[31] * -9.376056E-03;
netsum += inarray[32] * 0.1101889;
feature2[32] = 1 / (1 + exp(-netsum));

netsum = -3.34062E-03;
netsum += inarray[0] * 6.554157E-02;
netsum += inarray[1] * 0.2004918;
netsum += inarray[2] * 2.040677E-03;
netsum += inarray[3] * -0.1409248;
netsum += inarray[4] * 8.903714E-02;
netsum += inarray[5] * 0.4832121;
netsum += inarray[6] * 0.1188542;
netsum += inarray[7] * -0.2067401;
netsum += inarray[8] * -0.3984529;
netsum += inarray[9] * -0.1568604;
netsum += inarray[10] * -0.0385414;
netsum += inarray[11] * 8.957424E-02;
netsum += inarray[12] * -9.768099E-02;
netsum += inarray[13] * -3.784238E-02;
netsum += inarray[14] * -7.68872E-03;
netsum += inarray[15] * -1.002833E-02;
netsum += inarray[16] * 0.1242224;
netsum += inarray[17] * 5.571426E-02;
netsum += inarray[18] * 0.2149975;
netsum += inarray[19] * -0.4130803;
netsum += inarray[20] * -0.0451524;
netsum += inarray[21] * 0.3287497;
netsum += inarray[22] * 0.2796682;
netsum += inarray[23] * -3.541569E-02;
netsum += inarray[24] * -0.1946406;
netsum += inarray[25] * 6.054502E-02;
netsum += inarray[26] * 0.1861672;
```

```
netsum += inarray[27] * 7.808456E-02;
netsum += inarray[28] * -0.2864748;
netsum += inarray[29] * 0.2136261;
netsum += inarray[30] * 8.194526E-02;
netsum += inarray[31] * -0.2213097;
netsum += inarray[32] * -0.1077909;
feature2[33] = 1 / (1 + exp(-netsum));

netsum = -0.255922;
netsum += inarray[0] * 0.1260687;
netsum += inarray[1] * 0.1482843;
netsum += inarray[2] * -5.700107E-02;
netsum += inarray[3] * -0.175437;
netsum += inarray[4] * 0.2879411;
netsum += inarray[5] * -0.1840429;
netsum += inarray[6] * 0.2042107;
netsum += inarray[7] * 1.097267E-02;
netsum += inarray[8] * 0.2282462;
netsum += inarray[9] * -8.937116E-02;
netsum += inarray[10] * -2.944852E-02;
netsum += inarray[11] * 0.1414867;
netsum += inarray[12] * 0.2713032;
netsum += inarray[13] * -9.971763E-02;
netsum += inarray[14] * -7.193358E-02;
netsum += inarray[15] * -0.1119163;
netsum += inarray[16] * 4.504934E-03;
netsum += inarray[17] * 5.466465E-02;
netsum += inarray[18] * -2.626296E-02;
netsum += inarray[19] * 0.1583175;
netsum += inarray[20] * -0.2392287;
netsum += inarray[21] * -1.809105E-02;
netsum += inarray[22] * -0.1623578;
netsum += inarray[23] * 8.907694E-02;
netsum += inarray[24] * 4.621147E-02;
netsum += inarray[25] * 0.1517028;
netsum += inarray[26] * -0.1585366;
netsum += inarray[27] * -0.2341737;
netsum += inarray[28] * 0.1810547;
netsum += inarray[29] * -0.1812021;
netsum += inarray[30] * 0.2967823;
netsum += inarray[31] * -7.376265E-02;
netsum += inarray[32] * -0.2021585;
feature2[34] = 1 / (1 + exp(-netsum));

netsum = -0.1185325;
netsum += feature2[0] * 5.669532E-02;
netsum += feature2[1] * -0.5486844;
netsum += feature2[2] * 2.202653E-02;
netsum += feature2[3] * 8.283952E-02;
netsum += feature2[4] * -0.4856738;
netsum += feature2[5] * -9.785251E-02;
netsum += feature2[6] * -0.5676811;
netsum += feature2[7] * -0.2503108;
netsum += feature2[8] * 0.739545;
netsum += feature2[9] * 5.413911E-02;
netsum += feature2[10] * 0.2862896;
netsum += feature2[11] * 0.6443731;
netsum += feature2[12] * -0.1110446;
netsum += feature2[13] * -1.741772;
netsum += feature2[14] * 0.1380481;
netsum += feature2[15] * 0.8540484;
```

```
netsum += feature2[16] * 0.2201758;
netsum += feature2[17] * 0.6591824;
netsum += feature2[18] * 0.2710906;
netsum += feature2[19] * 0.4948081;
netsum += feature2[20] * 0.4218479;
netsum += feature2[21] * -0.2484195;
netsum += feature2[22] * -0.2045382;
netsum += feature2[23] * 0.9843633;
netsum += feature2[24] * 0.4215631;
netsum += feature2[25] * 1.226288;
netsum += feature2[26] * -0.8973132;
netsum += feature2[27] * -1.130871;
netsum += feature2[28] * 0.7940252;
netsum += feature2[29] * -0.5485248;
netsum += feature2[30] * -0.2591582;
netsum += feature2[31] * 2.136936E-02;
netsum += feature2[32] * -0.5458164;
netsum += feature2[33] * 0.6159741;
netsum += feature2[34] * -0.1191391;
outarray[0] = 1 / (1 + exp(-netsum));


outarray[0] = 410 * (outarray[0] - .1) / .8 + 302;
if (outarray[0]<302) outarray[0] = 302;
if (outarray[0]>712) outarray[0] = 712;
 c = outarray[0];

       return(c);

}
```