Water Distribution System Optimization using Metamodels

Darren Ross Broad

B.E. Civil/Env (Hons), B.Sc. (Ma. & Comp. Sc.)

Thesis submitted for the degree of

Doctor of Philosophy

The University of Adelaide

School of Civil, Environmental and Mining Engineering

September 2014

Contents

Abstract

Evolutionary Algorithms (EAs) have been shown to apply well to optimizing the design and operations of water distribution systems (WDS). Recent research in the field has focussed on improving existing EAs and developing new ones so as to obtain better solutions (closer to the global optimum) and/or find solutions more efficiently.

The primary aim of this research, however, has been to broaden the scope of optimization to include a number of the many factors that planning engineers need to consider when designing or planning the operations of WDS. Those factors considered here are (1) water quality criteria, (2) real-world, complex systems, and (3) the incorporation of data uncertainty.

Incorporating each of these factors independently increases computational run-time of EA-based optimization of an algorithm that is already computationally intensive compared to other (inferior) algorithms that have been used in WDS optimization. Water quality models tend to run slower than hydraulic models due to the shorter timestep that is required to ensure sufficient accuracy, and the need for extended period simulations thereby increasing the simulation duration. Real-world models run slower due to their size. Data uncertainty is typically accounted for through the use of Monte Carlo simulations, that add several orders of magnitude to the computational requirements of optimization.

Considering each of these factors together compounds the computational requirements to a point where it is impossible to optimize WDS using EAs in a reasonable amount of time. In this research metamodels have been used in place of simulation models within an EA to reduce this computational burden. A metamodel is a model of a model that runs much faster than the said model, but is still a high-fidelity approximation of it. The particular type of metamodel used in this research is an Artificial Neural Network (ANN) due to its theoretical capabilities and demonstrated effectiveness in water resources applications.

The use of metamodels to act as surrogates for complex simulation models is not a trivial task. Therefore, guidelines have been developed on how best to incorporate them into the WDS optimization process.

The overall metamodel-empowered, EA-based optimization algorithm developed in this research was applied to several case studies. Two small case studies, both variations of the New York Tunnels problem were studied for proof-of-concept purposes. They demonstrated that near globally-optimal solutions could still

be found using the metamodel-based approach, i.e. there was minimal compromise in the effectiveness of the EA-based approach. Two larger, real-world problems were also studied: Wallan (operations planning) and Pacific City (system augmentation). These last two case studies were key to demonstrating the power of using metamodels in that they enabled a computational speed-up of up to 1375 times (137,500%) compared to a non-metamodel approach. This speed-up includes factoring in the computational overheads of using metamodels, i.e. time to generate calibration data and calibrate the metamodels.

Declaration

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree.

I give consent to this copy of my thesis when deposited in the University Library, being made available for loan and photocopying, subject to the provisions of the Copyright Act 1968.

The author acknowledges that copyright of published works contained within this thesis resides with the copyright holder(s) of those works.

I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library catalogue and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

SIGNED:.....DATE:.....

List of Figures

List of Tables

List of Publications

The following is a list of the publications related to the research presented in this thesis:

Journal Papers:

Broad, D. R., Dandy, G. C., and Maier, H. R. (2005). "Water Distribution System Optimization Using Metamodels." *Journal of Water Resources Planning and Management - ASCE*, 131(3), 172-180.

Broad, D. R., Maier, H. R., and Dandy, G. C. (2010). "Optimal Operation of Complex Water Distribution Systems Using Metamodels." *Journal of Water Resources Planning and Management - ASCE*, 136(4), 433-443.

Broad, D. R., Dandy, G. C., and Maier, H. R., (2014). "Systematic approach to determining metamodel scope for risk-based optimization and its application to water distribution system design", *Environmental Modelling & Software,* Submitted.

Refereed Conference Paper:

Broad, D. R., Dandy, G. C., Maier, H. R., and Nixon, J. B. (2006). "Improving Metamodel-based Optimization of Water Distribution Systems with Local Search." *IEEE World Congress on Computational Intelligence, 16-21 July 2006*, Vancouver, BC, Canada, on CD-ROM.

Unrefereed Conference Papers:

Broad, D. R. (2004). "Incorporating Water Quality and Reliability into the Optimisation of Water Distribution Systems." *Fourth Postgraduate Student Conference of the CRC for Water Quality and Treatment, 14-16 April 2004*, Noosa, Queensland, 181-187.

Broad, D. R., Dandy, G. C., and Maier, H. R. (2004). "A Metamodeling Approach to Water Distribution System Optimization." *EWRI World Water and Environmental Resources Congress, 27 June - 1 July 2004*, Salt Lake City, Utah, USA, on CD-ROM.

Broad, D. R., Maier, H. R., Dandy, G. C., and Nixon, J. B. (2005). "Estimating Risk Measures for Water Distribution Systems using Metamodels." *EWRI World Water and Environmental Resources Congress, 15 - 19 May 2005*, Anchorage, Alaska, USA.

Broad, D. R. (2006). "Optimising Water Distribution Systems using Metamodels." *Fifth Postgraduate Student Conference of the CRC for Water Quality and Treatment, 10-13 July 2006*, Melbourne, Victoria.

Broad, D. R., Maier, H. R., Dandy, G. C., and Nixon, J. B. (2006). "Optimal Design of Water Distribution Systems including Water Quality and System Uncertainty." *8th Annual International Symposium on Water Distribution Systems Analysis, 27-30 August 2006*, Cincinnati, Ohio, USA, on CD-ROM.

Report:

Gibbs, M., Broad, D. R., Dandy, G. C., and Maier, H. R. (2010). "Decision Support Systems for Water Quality Optimisation." Water Quality Research Australia.

Presentation:

Broad, D. R. (2005). "Improving Water Distribution System Optimisation through the use of Metamodels", AWA Computer Modelling Special Interest Group.

Acknowledgments

Thanks firstly to my excellent, very patient, supervisors Prof. Graeme Dandy and Prof. Holger Maier. You have given me fantastic guidance, wisdom and encouragement over the years that will serve me well for the rest of my career. Thanks also to my industry supervisor, Dr. John Nixon for his guidance.

Thanks to the former Co-operative Research Centre for Water Quality and Treatment for providing financial support for my PhD. Thanks to Greg Ryan (formerly with South East Water) and Asoka Jayaratne (Yarra Valley Water) for their assistance in developing a real-world case study. Thanks to Chris Saliba for collating the necessary data for the Wallan case study and the guided tour of the area.

Thanks also to the federal government for providing an Australian Postgraduate Award.

Thanks to fellow postgrads Matt Gibbs and Rob May for all their technical help along the way. Thanks to my fellow postgraduate students (especially Nicole Arbon, Joe Davis, Matt Haskett, Greer Humphrey, Kylie Hyde, Pedro Lee, Michael Leonard, Dalius Misiunas, Steve Need, Jakin Ravalico, Mark Rebentrost, Tim Rowan, Mark Stephens, Jerry Vaculik, John Vitkovsky, Julian Whiting, Craig Willis, Aaron Zecchin, Matt and Rob) with whom I shared many interesting discussions, Friday night drinks, whinge sessions, and if memory serves correctly 10 ACPGNAPCs.

Thanks to my colleagues at Optimatics for their encouragement and helping me significantly in my coding skills.

Thanks to my friends who frequently asked the dreaded question for any decade-long student: "How's the PhD going?" Thanks especially to David McIver and Gerhard Bartodziej; you made me think about this PhD even when I didn't want to, and that probably helped me get over the line.

Thanks to my family for their love and support, especially my Mum for financial support and letting me move back home… twice.

Thanks to my good mate Ryan Ogilvy who was a great supporter and listener.

Finally, thanks to God who made me, forgave me, sustains me and gives me purpose.

Soli Deo gloria

Water Distribution System Optimization using Metamodels

Darren Ross Broad

Chapter 1

Background and

Publications Overview

"For a thousand years in thy sight are but as yesterday..."

Psalm 90:4a (KJV)

1.1 Introduction

Research into water distribution system (WDS) optimization increased significantly in the mid 1990s with the first application of Genetic Algorithms (GAs) to the problem (Simpson et al. 1994). It spawned a wave of research in applying what have come to be known as Evolutionary Algorithms (EAs) to WDSs. Many variations and improvements to EAs were developed, and new algorithms produced, all of which were demonstrated to perform well on WDS optimization (Dandy et al. 1996, Savic and Walters 1997, Cunha and Sousa 1999, Eusuff and Lansey 2003, Maier et al. 2003, Geem 2006).

EAs became more popular compared to previously used methods (e.g. linear programming, gradient search and enumeration) for the following reasons: they were simulation model based, they did not require complex gradients to be calculated, they did not require simplifications to the problem, they were less prone to becoming trapped in local optima, and they were easy to use (Simpson et al. 1996).

In the face of intensive research in the field, Walski (2001) was critical that WDS optimization was the wrong paradigm with regard to WDS planning. Walski's criticisms focused primarily on the fact that optimization omits many factors that design engineers/planners need to consider when designing a WDS in practice. That is, the problems most researchers considered were too simplistic. These factors include (1) the inclusion of water quality criteria to ensure the health of a utility's customers; (2) the ability to apply a planning paradigm to WDSs of any size (many researchers focussed on small, academic case studies, whereas larger real-world case studies, where optimization has greater potential benefits, bring a range of difficulties

to the design process); and (3) the need for robust designs, given knowledge of key design criteria (e.g. future demand), is imperfect.

To achieve broader acceptance of WDS optimization by industry, more of these aspects of realism must be considered by researchers. Consequently, this research has focussed on three ways of better incorporating these aspects of realism into the EA-based optimization of WDSs, as captured in the following aim.

Aim #1 of this research is to incorporate three aspects of realism into the optimization of WDS, namely (1) water quality criteria; (2) real-world systems; and (3) data uncertainty. It should be noted that other aspects of improving the realism with which the EA-based optimization of WDS is carried out, such as the inclusion of multiple competing objectives (e.g. Halhal et al. 1997, Kapelan et al. 2005) have not been considered in this research.

The one negative aspect of EAs compared with other optimization algorithms is that they are more computationally intensive. Adding more aspects of realism to the problem formulation dramatically increases this run time even more. And while computers have become faster and distributed computing has helped, currently it is not possible to optimize a WDS with an EA that takes all aspects into account that a planning engineer would consider in a reasonable amount of computing time.

This thesis takes a step towards enabling this goal to be achieved through the use of metamodels (Blanning, 1975, Razavi et al. 2012). A metamodel is a simplified model of a complex simulation model that solves much faster than the simulation model it approximates. The purpose of metamodelling is to reduce computational

time and hence it is very useful in applications of repetitive usage, such as EA-based optimization. Metamodel usage is non-trivial which leads to aim #2of this research.

Aim #2 of this research is to develop a robust methodology for the use of metamodels in WDS optimization.

1.2 Publications

This thesis is comprised of four publications. Their contribution to the body of knowledge in the research field can be most clearly presented with reference to the two broad aims of the research presented in Section 1.1.

The papers present significant advances in the incorporation of important aspects for WDS optimization. These are outlined in Section 1.2.1. Those advances are only facilitated through the use of metamodelling and significant advances in establishing a robust framework within which metamodels can be used for WDS optimization have also been made. These are summarised in Section 1.2.2.

1.2.1 Contributions to WDS Optimization

The contribution of the four papers to broadening the aspects considered in WDS optimization is presented in Table 1.1.

Table 1.1 Contributions to WDS optimization by publication.

Aspect of WDS Optimization	Publication			
	1	2	3	4
Water Quality	✓	✓	✓	✓
Real World Models			✓	✓
Data Uncertainty				✓

Publications 1 and 2 both include water quality criteria via modelling of disinfection parameters (chlorine dosing) and demonstrate the ability to optimize WDS with these criteria (they differ in their contribution to metamodelling knowledge; see Section 1.2.2.).

Publication 3 builds on the previous publications in that it also includes water quality criteria. Its main contribution is that it also includes an analysis of the additional factors that need to be considered when optimizing real world problems. A key outcome is that real world scale WDS problems that include water quality criteria can now be optimized (by using metamodels) whereas previously, as far as the author is aware, they could not.

Publication 4 also includes water quality criteria and the application to real world models, but builds on the previous publications by the addition of a way of accounting for key sources of data uncertainty (i.e. future demand, pipe roughness coefficients and chlorine decay rate). The paper presents the first demonstration of WDS optimization that includes water quality criteria, accounts for data uncertainty and applies it to real world models.

1.2.2 Contributions to the Development of a Metamodelling Framework for WDS Optimization

Several contributions have been made in establishing a framework within which metamodels may be used for WDS optimization. These contributions, found in the four publications, can best be presented when considering the steps involved in developing and using a metamodel for optimization.

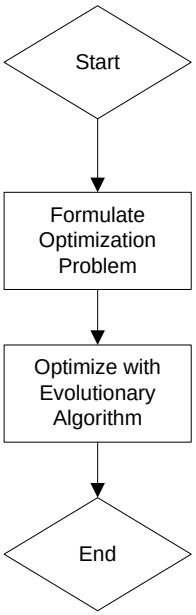


Figure 1-1 Traditional, or non-metamodel-based optimization procedure

Figure 1.1 shows the high level steps involved in EA-based optimization without the use of metamodels. In contrast, Figure 1-2 shows how the approach differs when metamodels are used. It is clear that when metamodels are not used, the optimization process is very straightforward; the optimization problem is formulated (decision variables, constraints and the objective function are defined), then the problem is optimized.

In contrast, the procedure when metamodels are used incorporates several additional steps. These are presented here in summary form, with references to where further details are laid out in the publications:

Step 1: Formulate Optimization Problem: This involves determining decision variables, constraints and data required to optimize a WDS. There are no differences in this step between the metamodel and non-metamodel scenarios.

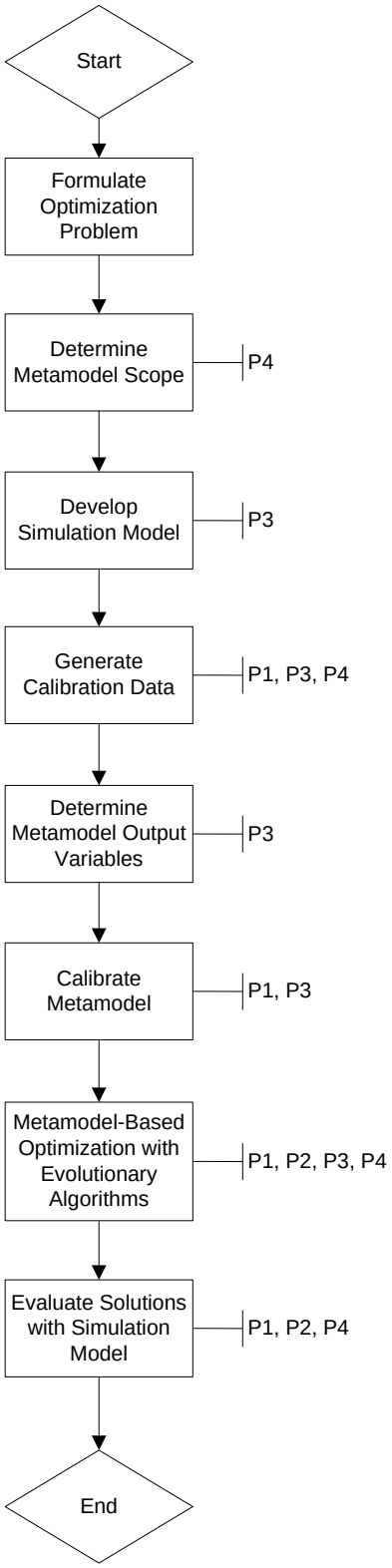


Figure 1-2 Metamodel-based optimization procedure with references to relevant publications for further detail.

Step 2: Determine Metamodel Scope: A systematic process was developed for determining the most appropriate scope of a metamodel; that is, which intermediate calculation steps of calculating a fitness function should be replaced with a metamodel. The purpose of this process is to ensure a metamodel replaces the most computationally intensive calculation steps, and the calculation steps whose structure is more easily approximated by a metamodel. A generic process was developed that may be used in any metamodelling application and the process was applied to the risk-based optimization of WDS using EAs (Publication 4).

Step 3: Prepare Simulation Model: Some key considerations need to be made when using metamodels for WDS optimization, especially when applying them to larger systems. One of the contributions of Publication 3 is a discussion of the considerations pertaining to preparing the simulation model so that it can be used in metamodel-based optimization.

Step 4: Generate Calibration Data: The simulation model is used to generate data that are needed to calibrate the metamodel. A general approach that is applicable to WDS design is presented in Publication 1. Publication 3 includes adaptations to this approach for WDS operations, and Publication 4 includes further adaptations to account for data uncertainty.

Step 5: Determine Metamodel Output Variables: The *types* of variables for which the metamodel needs to act as a surrogate are determined by the optimization problem (e.g. pressure and chlorine concentrations). However, modelling these variables at each node in the WDS system is computationally expensive and is not required. Publication 3 includes a numerical procedure to determine the minimum

number of nodes (output variables) for the metamodel, such that there is no change to the global optimum and fitness landscape compared to the traditional (non-metamodel) approach.

Step 6: Calibrate Metamodel: Each paper in this thesis uses Artificial Neural Networks (ANNs) as the type of metamodel. Publication 1 contains the justification for using ANNs, as well as a recommended general structure and calibration method for using them. Publication 3 contains a minor modification to the metamodel structure for operations optimization.

Step 7: Metamodel-based Optimization with Evolutionary Algorithms: Each paper in this thesis used EAs coupled with ANN metamodels. Publications 1 and 2 use metamodels to check constraint violations. Publication 3 uses metamodels for the same purpose, as well as for evaluating chlorine dosing costs and energy consumption by pumps. Publication 4 demonstrates the full power of metamodels by using them in the risk-based optimization of WDS. In that case, the metamodels are used to calculate pressure and chlorine residuals that are used to calculate risk metrics (reliability and vulnerability) within a Monte Carlo Simulation; the risk metrics are then used as constraints in the EA.

Step 8: Evaluate Solutions with Simulation Model: A metamodel acts as a high-fidelity approximation to a simulation model, however, it is not a perfect approximation. Therefore, after running an EA, some solutions need to be checked with the original simulation model. Publication 1 presents an algorithm that includes several solution-checking steps. Publication 2 presents an evaluation of a range of different algorithms that were developed, or selected, specifically for WDS

optimization. Publication 4 presents a modification to the original algorithm that recognises that, in practical situations, there may be a limited computational budget. In light of this, the paper examines how best to use the limited fitness evaluations.

1.2.3 Summary

Each publication is presented in the following four chapters. The contributions to knowledge that each paper provides is given in detail in Sections 1.21-1.2.2. For simplicity, each chapter has been renamed to reflect the main contributions of each paper, as outlined in Table 1.2.

Table 1.2. Publications, chapters and their main focus.

Publication	Chapter	Main Focus
1	2	Water Quality
2	3	Local Search
3	4	Complex Hydraulic Systems
4	5	Data Uncertainty

Chapter 2

Publication 1: Water Quality

"...the angel showed me the river of the water of life,

as clear as crystal, flowing from the throne of God..."

Revelation 22:1 (NIV)

Statement of Authorship

Title of Paper	Water Distribution System Optimization Using Metamodels.
Publication Status	Published
Publication Details	Broad, D. R., Dandy, G. C., and Maier, H. R. (2005). "Water Distribution System Optimization Using Metamodels." *Journal of Water Resources Planning and Management - ASCE*, 131(3), 172-180.

Author Contributions

Name of Co-Author	Darren Broad		
Contribution to the Paper	Conceptual and theoretical development, interpretation and analysis of results, manuscript preparation and corresponding author.		
Signature		Date	

Name of Co-Author	Graeme Dandy		
Contribution to the Paper	Project supervision and review of manuscript.		
Signature		Date	

Name of Co-Author	Holger Maier		
Contribution to the Paper	Project supervision and review of manuscript.		
Signature		Date	

Although the manuscript has been reformatted in accordance University guidelines, and sections have been renumbered for inclusion within this thesis, the paper is otherwise presented herein as published.

Abstract

Genetic Algorithms (GAs) have been shown to apply well to optimizing the design and operations of water distribution systems (WDS). The objective has usually been to minimize cost, subject to hydraulic constraints, such as satisfying minimum pressure. More recently, the focus of optimization has expanded to include water quality concerns. This added complexity significantly increases computational requirements of optimization. Considerable savings in computer time can be achieved by using a technique known as metamodeling. A metamodel is a surrogate, or substitute for a complex simulation model. In this research, a metamodeling approach is used to optimize a water distribution design problem that includes water quality. The type of metamodels used are Artificial Neural Networks (ANNs), as they are capable of approximating the non-linear functions that govern flow and chlorine decay in a WDS. The ANNs were calibrated so as to provide a good approximation to the simulation model. In addition, two techniques are presented to improve the ability of metamodels to find the same optimal solution as the simulation model. Large savings in computer time occurred from training the ANNs to approximate chlorine concentrations (approximately 700 times faster than the simulation model) while still finding the optimal solution.

2.1 Introduction

Many decisions need to be made with regard to the design and operation of Water Distribution Systems (WDS). These decisions are generally aided by the use of a simulation model, such as EPANET. The simulation model enables decision makers to examine the effects of their decisions prior to implementation. Traditionally, several sets of decision variables or solutions may be evaluated using the simulation model, with the best set selected for implementation. Optimization algorithms have an advantage in that the process of simulating the solutions can be automated with many thousand solutions evaluated in such a way as to guide the search towards the optimum.

2.2 Literature Review

WDS Optimization has existed as a research field for over 30 years and can be broadly classified into two application areas, namely design and operations. The focus of this paper is on the optimal design of WDS. Design can include, for example, determining pipe diameters and locations of chlorine booster stations.

Since Simpson et al. (1994), Genetic Algorithms (GAs) have been applied extensively to optimize WDS for hydraulic criteria. The main advantages of GAs are that they use a population of evolving solutions and identify several solutions from which the decision maker can select, rather than a single optimum. The main disadvantage lies in the high computational intensity.

Improvements have been made in the GA (Dandy et al. 1996; Walters et al. 1999) and there have been improvements in the usefulness of optimization as a tool for designing WDS through the addition of non-hydraulic constraints. In practice,

water authorities need to satisfy water quality criteria in addition to meeting hydraulic constraints while minimizing cost. Upper and lower bounds on disinfection are needed to avoid taste and odor complaints and to ensure there is no microbial contamination, respectively. Dandy and Hewitson (2000) incorporated water quality issues into optimizing the design of WDS with a GA. The key finding was that when water quality constraints were included, the optimum solution was more costly than that obtained when only hydraulics were considered. However, water quality-based optimization has a much higher computational burden, relative to hydraulics, due to the shorter computational time-step of the simulation model and the need to run an extended period simulation. Therefore, for water quality issues to be incorporated into optimization, methods to reduce this computational burden need to be developed.

The technique proposed in this paper to reduce the computational intensity of water quality-based optimization is known as metamodeling. A metamodel, first proposed by Blanning (1975), is a model of a simulation model. The metamodel serves as a surrogate, or substitute, for the more complex and computationally expensive simulation model, which is EPANET in the case of WDS optimization. While it does take time to develop metamodels, this is generally offset by the considerable time savings arising when they are linked with an iterative algorithm that requires them to be run many times, such as a GA. There are several different types of metamodel that can be constructed, including regression models and artificial neural networks (ANNs). In this paper, the focus is on using ANNs as a metamodel for the complex WDS simulation models. The advantage that ANNs have over regression metamodels is that they can represent complex, non-linear functions without the

need to pre-determine the form of the model (eg. linear, polynomial) (Leshno et al. 1993).

Metamodels have been used in a wide variety of applications, including calibrating WDS (Lingireddy and Ormsbee 1998), modeling of chemical reactors (Kalagnanam and Diwekar 1997) and modeling of aircraft operation (Meckesheimer et al. 2002). Of particular interest is the work of Aly and Peralta (1999) and Johnson and Rogers (2000), who used ANN metamodels as approximations to complex groundwater models and subsequently used them in place of the simulation model in an optimization framework. After training the ANN, Aly and Peralta (1999) accepted it as an accurate approximation to the simulation model and used it for optimization. Johnson and Rogers (2000) improved on this by checking the optimal solution found by the ANN linked to an optimization algorithm with the simulation model. However, because it is unlikely that the ANN could provide a perfect approximation to the simulation model, it is insufficient to simply check one solution for feasibility. If it turned out that that solution was infeasible it would compromise the entire metamodeling process. Therefore, optimization runs that utilize metamodels should incorporate a broader method of checking feasibility of the solutions obtained with the original simulation model. Such a method is presented in this paper and applied to a benchmark problem from the literature.

2.3 Objectives

This paper presents a methodology for developing ANN metamodels for WDS for the purpose of reducing computational runtimes for the optimal design of WDS that include hydraulic and water quality constraints. In addition, two techniques are

presented to improve the performance of ANN metamodels in finding optimal solutions. These include evaluating select solutions with EPANET during optimization and adjusting constraints slightly to account for small errors in the metamodel.

2.4 Optimization Approach

In general terms, a water quality and hydraulics based optimization problem aims to minimize cost, such that constraints on pressures and chlorine concentrations at demand nodes are within certain bounds. The optimal design formulation used in this paper is given by Eq. (2.1) to Eq. (2.3).

$$\min z(\phi) = \sum_{i=1}^{n} UC_{\phi_i} L_i \tag{2.1}$$

$$P_{j-min} \leq P_j \leq P_{j-max}, j = 1, \dots, m \tag{2.2}$$

$$C_{j-min} \leq C_j \leq C_{j-max}, j = 1, \dots, m \tag{2.2}$$

Where $z(\phi)$ is the cost of design ϕ; UC_{ϕ_i} is the cost per unit length of pipe i for design ϕ; L_i is the length of pipe i; P_j, $P_{j\text{-}min}$ and $P_{j\text{-}max}$ are the pressure at node j and the minimum and maximum allowable pressures, respectively; C_j, $C_{j\text{-}min}$ and $C_{j\text{-}max}$ are the residual chlorine concentration at node j and the minimum and maximum allowable chlorine residuals, respectively; m is the number of nodes (or critical nodes) in the WDS; and n is the number of pipe segments. The constraints given by Eq. (2.2) and Eq. (2.3) are specifically relevant to the GA. However, there are also constraints regarding continuity, headloss and chlorine decay. While these do need to be satisfied, they are internal to the simulation model and thus are not presented here. To optimize this with a GA, constraints are converted into penalty costs, as shown in Eq. (2.4).

$$\min z(\phi) = \sum_{i=1}^{n} UC_{\phi_i} L_i + PC_1 + PC_2 \tag{2.4}$$

Where PC_1 and PC_2 are the penalty costs for pressure head and chlorine residual, respectively given by the following:

$$PC_1 = \max_j \left[\max\{0, (P_{j-min} - P_j), (P_j - P_{j-max})\}\right] PM_1 \tag{2.5}$$

$$PC_2 = \max_j \left[\max\{0, (C_{j-min} - C_j), (C_j - C_{j-max})\}\right] PM_2 \tag{2.6}$$

Where PM_1 is the penalty multiplier for pressure head ($/m) and PM_2 is the penalty multiplier for chlorine residual ($/mg/L).

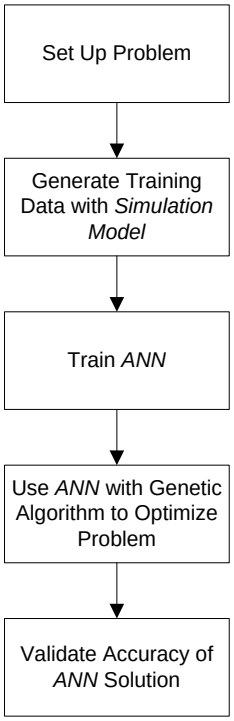
2.5 Metamodel Development

The processes that need to be followed in developing an ANN metamodel for use in place of a simulation model for optimization are shown in Figure 2-1. While the purpose of using a metamodel is to reduce computer runtime, the extra steps required in the development phase of the metamodel reduce the time savings obtained.

In order to find optimal or near-optimal solutions when using a GA, a simulation model is needed to check whether the constraints are violated for a given set of decision variables (and if so, a penalty cost must be applied). Put simply, the purpose of the simulation model is to model the constrained variables as a function of decision variables. Therefore, a metamodel in place of a simulation model will need to do the same. For an ANN, the structure would entail decision variables at the input layer and constrained variables at the output layer. An example of an ANN that is used as a surrogate for a simulation model in a simple four-pipe, four-node, optimal design

problem is illustrated in Figure 2-2. In this example, the optimal solution would need to satisfy minimum pressure and minimum chlorine concentrations at the extreme demand node of the network, which would be dependent upon pipe diameters and the chlorine dosing rate. Hence the inputs and outputs of the ANN are as shown in Figure 2-2.

(a)



(b)

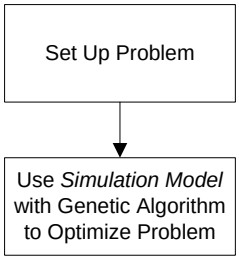


Figure 2-1. Steps in optimizing a water distribution system, (a) ANN metamodel approach, (b) simulation model approach.

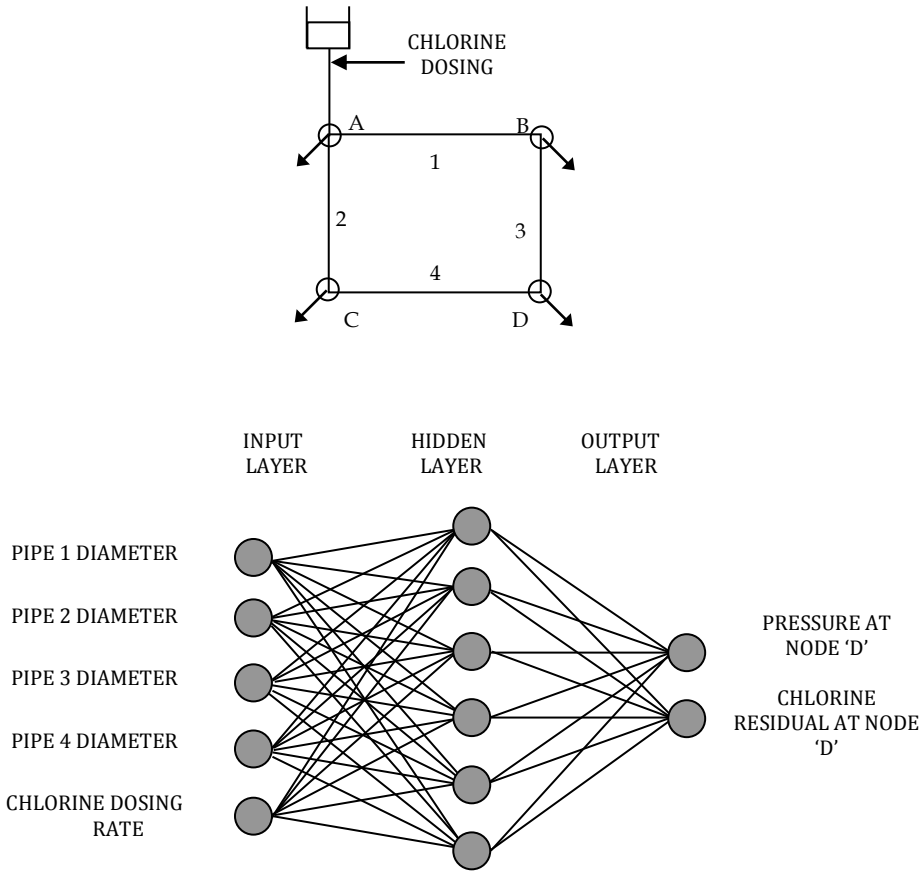


Figure 2-2. A simple WDS and its ANN metamodel.

There are two possible methods for obtaining data to train the ANN. Given that an ANN metamodel is trained using synthetic data generated from a simulation model, there will not be any noise in the data (Johnson and Rogers 2000). Noisy data can result in over-training of the ANN, leading to poor generalisation ability when presented with new data unseen in the training process. One method is to generate data initially across the entire search-space and then train the ANN (Johnson and Rogers 2000), while another method is to train the ANN while running the GA, and use the solutions the GA obtains to periodically re-calibrate the ANN (Lingireddy and Ormsbee 1998). While further study is needed to determine the more appropriate method, the former approach was used in this study. This is because GAs continually

explore different regions of the search-space. Consequently, there is the danger in the latter approach that the metamodel is only accurate in a small region of the search-space and will perform poorly in freshly explored regions, which may in fact contain the global optimum.

In order to develop a representative ANN, training data need to be generated from a range of different values and for different types of variables, depending on the optimization situation being considered. In a design situation, values for each pipe diameter and dosing rate should be sampled. Therefore the technique for generating training data that was used in this research was uniform random sampling. It is important to ensure the sampled data cover the whole search space because, while ANNs can interpolate between solutions, they cannot extrapolate well. Therefore, in addition to the randomly sampled data, points corresponding to the extremes of the possible values of the output variables were also sampled. In a design example, maximum pressures correspond to maximum possible diameters, while minimum pressures correspond to the smallest pipe diameters. For water quality, maximum chlorine residuals correspond to a solution consisting of the maximum dosing rates and minimum pipe diameters (for small detention times), while minimum residuals correspond to minimum dosing rates and maximum pipe diameters.

2.6 Optimization with Metamodels

Where a trained metamodel is used in place of EPANET in a GA run, penalty costs are calculated with the following equations:

$$PC_1 = \max_j \left[\max\{0, (P_{j-min} - \widetilde{P}_j), (\widetilde{P}_j - P_{j-max})\}\right] PM_1 \quad (2.7)$$

$$PC_2 = \max_j \left[\max\{0, (C_{j-min} - \widetilde{C}_j), (\widetilde{C}_j - C_{j-max})\}\right] PM_2 \quad (2.8)$$

Where $\widetilde{P}_j$ and $\widetilde{C}_j$ replace P_j and C_j from Eq. (2.5) and Eq. (2.6) and are the nodal pressure head and chlorine residual approximations calculated by the ANN, respectively. In order to evaluate the effectiveness of this proposed approach, a comparison was made with the solutions found with an ANN linked to a GA (henceforth referred to as ANN-GA) and the optimal solutions found by EPANET linked to a GA (EPANET-GA). This comparison was made for this paper, however in practice, one would not be able to compare the two approaches. Metamodels should only be used where time constraints prohibit the possibility of optimizing a problem with a simulation model.

2.6.1 Checking Solutions with the Simulation Model

In order to ensure feasibility, it is important to evaluate solutions found by the ANN-GA with EPANET. This is required since it is unlikely that the ANN would be able to provide a perfect approximation to the EPANET model. However, simply checking the single solution to which the ANN-GA converges would not be adequate, because one could not be certain that this is indeed the optimum or even feasible, due to errors in the ANN. Therefore, it is proposed to use a three-stage approach for evaluating select solutions with EPANET.

The first stage involves keeping track of several of the top solutions (rather than the single best) as the ANN-GA progresses and then evaluating these solutions with EPANET after the GA has converged. The logic behind this approach is that the solution to which the ANN-GA converges may actually be slightly infeasible when

modeled with EPANET. However, slightly more expensive solutions than the one to which the ANN-GA converges are more likely to be feasible. Hence the top solutions found using the ANN-GA will all be deemed feasible by the ANN but may consist of both feasible and infeasible solutions according to EPANET.

Figure 2-3a illustrates the usefulness of keeping track of a number of the top solutions when attempting to find the optimal solution. The real optimal solution (according to EPANET) is point A, however point B is the optimal solution according to the ANN. Point C will be one of those top solutions that are tracked because it is a good, but sub-optimal solution according to the ANN. After the ANN-GA has converged, points C and B will be evaluated with EPANET, with real costs of A and D, respectively. Therefore the optimal solution will be found with the ANN-GA that checks the top few solutions with EPANET.

The second stage is to conduct a local search after the ANN-GA has converged. This stage is proposed because the solution to which the ANN-GA converges may be sub-optimal. Hence a local search may find a slightly better solution, given that it will begin with a very good starting position. The local search used in this research consists of sequentially selecting each decision variable and reducing it by one value to a less expensive solution. If that solution is feasible according to EPANET, then a second reduction of that decision variable is performed. However, if the solution is infeasible, that variable in increased again and the process moves onto the next decision variable. This process continues until all decision variables have been reduced to the point where there is no further improvement. An example of the benefit a local search provides after the ANN-GA has converged is shown in Figure 2-3b. The real optimum is at point E, while the ANN-GA will converge to point F. A

local gradient search conducted with EPANET will commence from point G. Hence the actual optimum at point E should be found by a local search conducted after the ANN-GA has converged.

The third and final stage is to evaluate each new best solution found by the ANN-GA with EPANET. This is needed in conjunction with tracking the top few solutions because errors in the ANN (with respect to EPANET) may be large enough such that all of the top few solutions are actually infeasible when checked by EPANET. Without this stage of the proposed process, the only way to obtain feasible solutions would be with a local search. Hence by including an extra stage of evaluating each new best solution with EPANET, it will be more likely that the actual optimum will be found. Figure 2-3c illustrates two separate sections of a possible search space. The optimal solution is at point H, whereas the ANN-GA will converge to point J. However, considering the way in which a GA operates, in that it can escape local optima, point K may be found by the ANN-GA while searching for point J. Hence, when this solution is evaluated with EPANET, the optimal solution (H) will be found. The benefit of this third stage is further highlighted when it is used in conjunction with a local search. Point K itself does not need to be found, instead, point L (or any other point between the two) could be found by the ANN-GA before it converges on point J. Then, when the local search is conducted with EPANET from point M (point L for the ANN), it should find the optimal solution at point H.

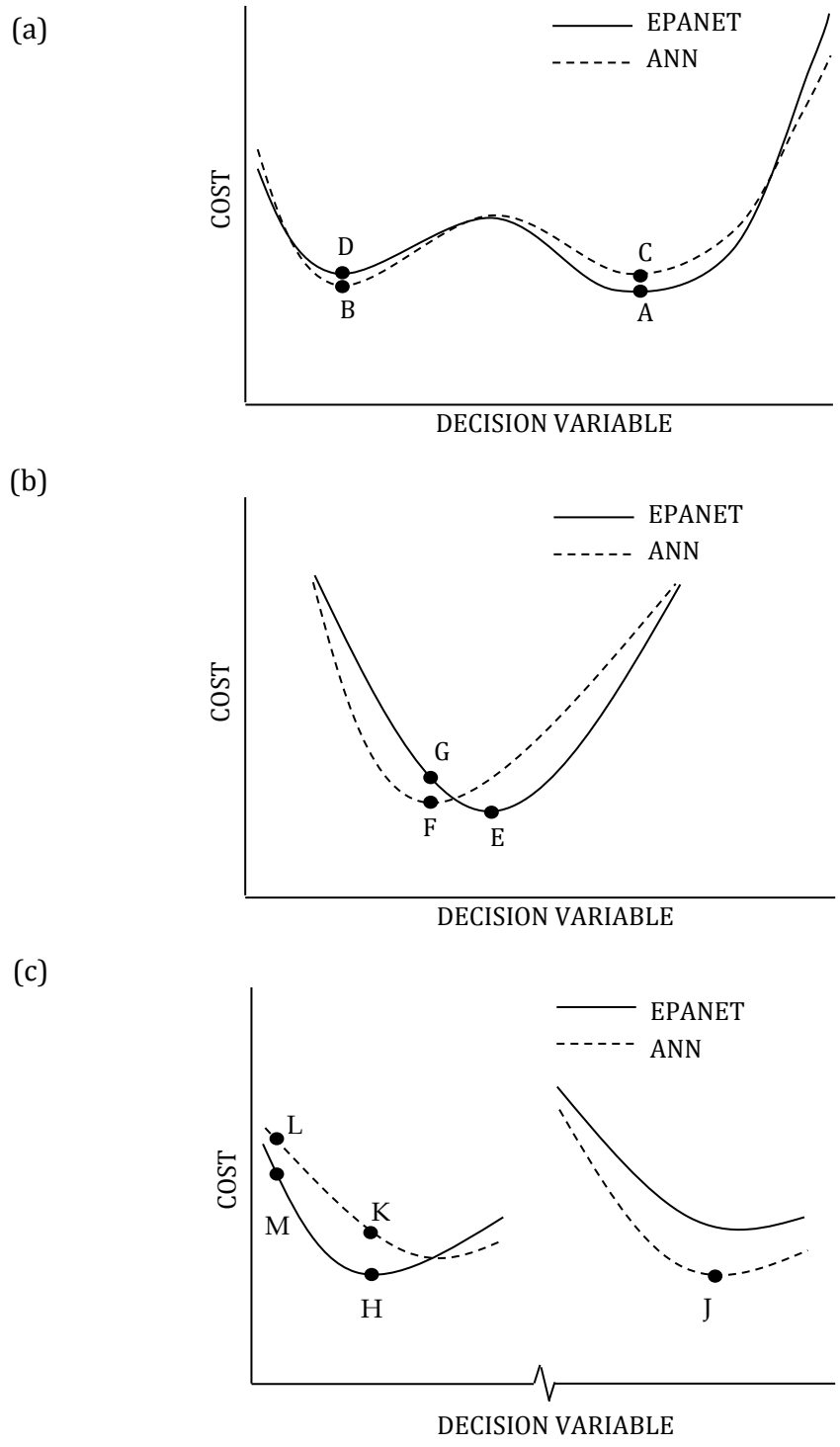


Figure 2-3. Strategies of obtaining the optimal solution through the evaluation of select solutions with EPANET, (a) top few solutions, (b) local search, (c) new optimal solutions.

In summary, each new best solution is checked for feasibility with EPANET as the ANN-GA runs. After convergence has been achieved, the top few solutions found by the ANN-GA are checked for feasibility by EPANET. Finally a local search is conducted with EPANET around the best feasible solution found during the first two stages. Note that this three-stage approach uses Eq. (2.5) and Eq. (2.6) to calculate penalty costs, whereas the ANN-GA uses Eq. (2.7) and Eq. (2.8).

2.6.2 Constraint Adjustment

As mentioned previously, the metamodel is unlikely to be able to provide a perfect representation of the simulation model. A proposed method of combating this issue is to adjust the constraints used by the GA while linked to the ANN. For example, if the ANN under-estimates pressure at the optimal solution, it would be necessary to relax the constraints somewhat to avoid the ANN-GA converging to a sub-optimal solution. Similarly, if the ANN over-estimates pressure, the constraints should be tightened so that the ANN-GA does not converge to an infeasible solution. Therefore, the penalty cost functions become the following.

$$PC_1 = \max_j \left[\max\left\{0, \left(\tilde{P}_{j-min} - \tilde{P}_j\right), \left(\tilde{P}_j - \tilde{P}_{j-max}\right)\right\}\right] PM_1 \tag{2.9}$$

$$PC_2 = \max_j \left[\max\left\{0, \left(\tilde{C}_{j-min} - \tilde{C}_j\right), \left(\tilde{C}_j - \tilde{C}_{j-max}\right)\right\}\right] PM_2 \tag{2.10}$$

Where $\tilde{P}_{j-min}$, $\tilde{P}_{j-max}$, $\tilde{C}_{j-min}$and $\tilde{C}_{j-max}$ are the adjusted acceptable minimum and maximum pressure and chlorine residuals, respectively.

An example of pressure at a critical node as a function of the diameter of one pipe is shown in Figure 2-4. The minimum acceptable pressure constraint is at P_{min}, meaning that solutions with diameters smaller than D_{min} will incur penalty costs. The solution at D_{min} is also the global optimum. Now consider an ANN approximation to an

EPANET model that over-estimates pressure in the region of the optimum. The ANN-GA will converge to a cheaper solution, due to smaller pipe diameters, at D^o, and will only incur penalty costs for solutions below D^o. However, this solution, when checked with EPANET, is actually infeasible. Conversely, consider an ANN that under-estimates pressure in the region around the optimum. In this case the ANN-GA will apply penalty costs to all solutions below D^u. Hence the solution D_{min} will erroneously have a penalty cost added to it and the ANN-GA will converge to a sub-optimal solution.

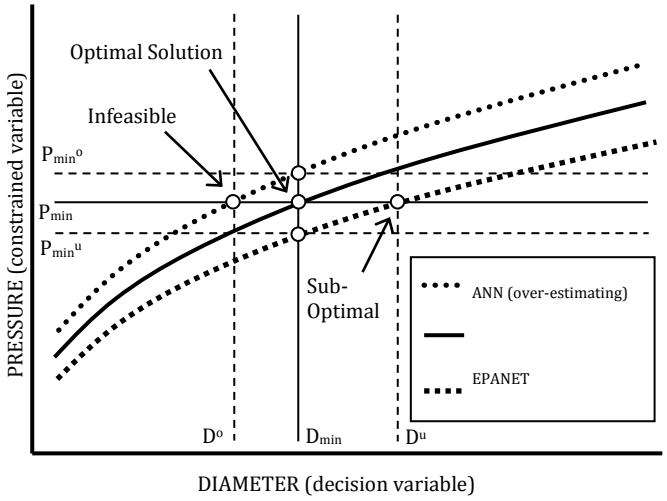


Figure 2-4. ANN approximations to an EPANET model.

Hence over-estimating and under-estimating ANNs will converge to infeasible and sub-optimal solutions, respectively, when the same constraints are used (P_{min}) for the ANN-GA as for the EPANET-GA. Therefore, it is proposed to adjust the constraints slightly so that penalty costs begin to be added at the same solution for the ANN as for EPANET. Subsequently, the ANN-GA will be more likely to converge to the same solution as EPANET-GA. It should be noted that every time a solution is checked using EPANET, the original constraints are still used. An ANN that under-estimates pressure

needs the constraints to be relaxed slightly to P_{min}^{u} so that the ANN-GA will converge to D_{min}, while an ANN that over-estimates pressure should have slightly tighter constraints at P_{min}^{o}. Eq. (2.11) explicitly shows how minimum pressure constraints are adjusted in order to achieve tighter or more relaxed constraints.

$$\tilde{P}_{i-min} = (1-\propto RMSE_i)P_{i-min} \tag{2.11}$$

Where $\tilde{P}_{i-min}$ is the relaxed minimum pressure for node *i*, $RMSE_i$ is the root mean squared error from the validation set for node *i*, and α is a constant, valid in the interval $[-\infty,+\infty]$. It should be noted that negative values of α will result in tighter constraints, while positive values of α will result in relaxed constraints. Without any knowledge of the size of the error of the ANN with respect to EPANET at the optimum solution, the RMS error is used since it provides an average error of the ANN across the entire search space. However, this is only an average error and the ANN may be more or less accurate than this at the optimum, hence the use of a constant term, α. Also, the ANN could over-estimate pressure or chlorine residual in some regions of the search space and under-estimate it in others.

A problem arises is that in practice one would not know whether the ANN over or under-estimates pressure near the optimal solution, as the optimum itself is unknown. Given that the time taken to optimise the problem with the ANN-GA is significantly less than that taken with the EPANET-GA, several optimisation runs can be made. Therefore it is proposed that a range of values for α should be used, rather than a single value. As mentioned above, both positive and negative α values should be used, because in practice one would not know if the ANN-GA is converging to a solution larger or smaller than the optimum.

2.7 Case Study

The New York Tunnels problem was chosen as the case study on the basis that there has been considerable research conducted on it in the past and therefore the current best solution is probably close to, if not *the*, global optimum. Therefore, this enables the effectiveness of the proposed approach to be evaluated. The NYT problem is a WDS expansion problem, where the optimal set of diameters of pipe segments need to be determined such that pressure at all nodes is above a specified minimum for a given set of demands. Further details of the NYT problem can be found in Maier et al. (2003), which also contains the current best known solution of $38.64m when EPANET 2.0 is used as the hydraulic solver. In this study, the problem was adapted to include water quality. This was achieved by adding a decision variable that represents the chlorine dosing rate at the reservoir at the start of the WDS. Possible dosing levels ranged from 0.5mg/L to 2.5mg/L in increments of 0.1mg/L. A minimum chlorine concentration of 0.3 mg/L throughout the system was set as the water quality constraint and the chlorine decay rate was assumed to be 1.0 day^{-1}.

2.7.1 Analysis Conducted

With 21 different pipe segments, each with 16 possible diameters, and 21 possible chlorine dosing rates, there are 4.1×10^{26} possible solutions to the water quality-adapted New York Tunnels problem (NYT-WQ). Ten thousand randomly generated solutions were sampled from this space and used to train the ANN metamodels. From these solutions, four nodes were found to be critical in terms of minimum pressure surplus (16, 17, 19 and 20) and one node was critical in terms of minimum chlorine residual (17).

Two different metamodeling scenarios were used as surrogates for the NYT-WQ EPANET model. Both scenarios included five separate ANNs, each with a single output. They differed in that scenario B used a different randomly generated data set for training than scenario A. Different scenarios were chosen to determine the relative capabilities of each in approximating EPANET and also in finding the optimal solution.

In order to test the repeatability of results, three different ANNs were trained from the same training data for each scenario. This was done to determine whether initial weights in the ANNs affected the ability of the ANN-GA to find the optimal solution.

The back-propagation algorithm was used to train the ANNs. The stopping criterion for this training method is generally when the error in an independent test set is at a minimum. However, as the data used to calibrate the ANNs were generated with a simulation model and hence were not noisy, it was found that the error in the test set steadily decreased for a large number of iterations. Therefore an alternative stopping criterion of a fixed number of iterations was used.

Optimization was conducted using a Genetic Algorithm (GA) with integer coding, one-point crossover and a tournament size of two. The type of ANN used was the multi-layer Perceptron (MLP) with a single hidden layer, as this is sufficient for approximating any continuous function (Leshno et al. 1993). The values of all ANN and GA parameters were selected by trial-and-error to obtain the best performance without excessive computational time and are given in Tables 2.1 and 2.2, respectively.

Table 2.1. ANN Parameters

Parameter	Value
ANN Type	multi-layer perceptron
Transfer Function	sigmoid
No. Hidden Layers	1
No. Hidden Nodes	40
Learning Rate	0.3
Momentum Rate	0.5
Training Iterations per ANN	5000

To illustrate the relative benefits of checking certain solutions with EPANET, a comparison was conducted between combinations of the three stages mentioned previously. The comparison was made for both scenarios A and B. The different combinations of EPANET strategies used for each of the ANNs were as follows:

- Checking each new best solution found;
- Checking each new best solution and the top forty solutions;
- Checking each new best solution, and conducting a local search; and
- Checking each new best solution, the top forty solutions, and conducting a local search.

Table 2.2. GA Parameters

Parameter	Value
GA Type	integer
Population Size	400
Probability of Crossover	0.8
No. Crossovers per Pair	1
Probability of Bit-Wise Mutation	0.02
No. Generations	2000
Penalty Multiplier[a] [$/m] or [$/mg/L]	10^9

a: The penalty multipliers for pressure head and chlorine residual are measured in $/m and $/mg/L, respectively.

In evaluating the constraint adjustment technique, a range of values for α (see Eq. (2.11)) were used. The parameter α relates to the RMSE which only gives the average error of the ANN with respect to EPANET, whereas the error at the optimal solution (which is of most interest) could be higher or lower than the RMSE. Therefore, values of α in this study ranged from –5 to 5 in increments of 0.1, with 10 optimization runs conducted at each α value.

2.7.2 Metamodel Performance

The accuracy of each of the ANNs was evaluated for an independent validation set of 1000 data points, not used in training the ANNs. Infeasible data, as well as feasible data, were used to construct the metamodels. This is because when an ANN is linked to a GA it is equally important that the metamodel accurately approximates outputs in the feasible and infeasible regions to determine whether a constraint has been violated.

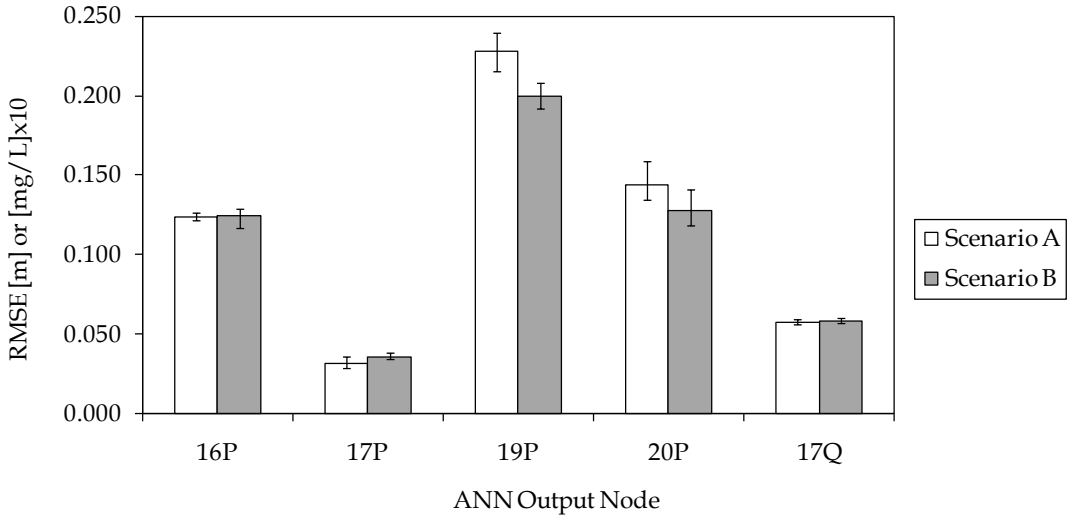


Figure 2-5. RMSE using two different training sets for the critical nodes of the NYT-WQ problem.

A summary of the accuracy of the ANNs is shown in Fig. 2.5. The average root-mean-squared error is given for all five outputs, with the error bars representing the minimum and maximum values across the three ANNs that were trained for each scenario. There is a fairly large difference between the errors for the pressure output nodes. Node 19 has a much larger error than node 17, for example. This is due to the fact that the training data for node 19 cover a broader range of values than the data for node 17, which is a feature specific to the NYT problem. With the exception of node 19, there is little difference between the RMS errors for ANNs trained with different data sets, when taking into account the range covered by the maximum and minimum bars. Therefore, it can be concluded that the randomness of initial weights in the ANNs has a greater influence on the final accuracy of the trained ANNs than the data set on which they are trained.

Overall, the performance of the ANN metamodels was very good. The accuracies of the ANNs are especially encouraging when one considers that the EPANET model itself is not a perfect representation of the actual WDS. The ANN metamodels gave a reasonable approximation to the EPANET model. Therefore the ANN metamodels can be used with confidence as surrogates for EPANET in an optimization problem.

2.7.3 Optimization Results – EPANET

While this research has used an adapted NYT problem (including water quality), actual costs of treatment have been neglected (as it is assumed capital costs of pipes to be far greater). The optimal solution of this problem, obtained with EPANET linked to a GA, was found to have the same cost ($38.64m) and pipe diameters as the ordinary NYT problem. A chlorine dosing rate of 1.7-2.5mg/L, as the additional decision variable, was found to satisfy the water quality constraint. The optimal solution was found three times from five GA runs. This result is contrary to the findings of Dandy and Hewitson (2000), who found that the optimal solution increased in cost when water quality constraints were included, for the NYT problem. However, this result would be specific to the formulation of the problem. Dandy and Hewitson (2000) proposed a social cost methodology based on the risk and cost of microbial infection due to insufficient disinfection.

2.7.4 Optimization Results – Metamodels

Table 2.3 compares the ability of each strategy in finding the optimal solution when using EPANET to check solutions from the ANN-GA. Simply checking each new best solution with EPANET as it is found by the ANN-GA is not sufficient, with only one of the ANNs of scenario A and none of the ANNs of scenario B finding the optimal

solution. Generally speaking, checking the top forty solutions, as well as each new best solution, provides little additional benefit by way of a better optimum. However, a local search appears to be a more useful strategy than checking the top forty solutions. As with the RMS error of the ANNs, it is clear that initial weights in the ANN have a big impact and affect the ability of the ANN-GA to find the optimal solution.

Table 2.3. Solutions found [$million] for the NYT-WQ problem with different methods of checking solutions with EPANET, with the current best-known solution in italics and the number of times it was found out of 10 runs in brackets.

Scenario	ANN Number	ANN-GA	Solutions Checked with EPANET			
			NB[a]	NB, TF[b]	NB, LS[c]	NB, TF, LS
A	1	37.95*	*38.64*(1)	*38.64*(6)	*38.64*(3)	*38.64*(5)
	2	38.09*	39.82	40.09	38.80	39.06
	3	39.85	39.85	39.85	38.80	38.80
B	1	41.07	40.69	41.07	38.80	38.80
	2	37.85*	39.28	39.60	*38.64*(1)	38.80
	3	42.27	42.27	42.27	39.21	39.21

a: NB: New Best. b: TF: Top Forty. c: LS: Local Search
* Infeasible when checked with EPANET

While the optimal solution of $38.64m was found at least once for each scenario, the repeatability of finding that solution is important. Each of the results presented in Table 2.3 are the minima from ten optimization runs. With all three stages of checking solutions with EPANET, the optimal solution was found for on five out of ten runs. This contrasts with scenario B, for which the optimal solution was only found once. Hence, while there was little difference between the RMSE for each scenario, there is much greater difference in their respective abilities to find the

optimal solution consistently. Also in Table 2.3, for comparison, are the solutions to which the ANN-GA converged. It can be seen that the ANN-GA sometimes converged to infeasible solutions, thus highlighting the need for a modified approach.

2.7.5 Optimization Results – Adjusted Constraints

The results presented here are for ANN #3 of scenario A, which is a metamodel that could not find the optimal solution without constraint adjustment, even with a local search, checking the top forty solutions and each new best solution with EPANET. The results for this ANN, which are shown in Fig. 6, indicate that constraints needed to be relaxed slightly for the optimum to be found, indicating the ANN was over-estimating pressure in the region around the optimum. This result highlights the success of the strategy of constraint adjustment in order to find the optimal solution. The results also highlight the importance of trying a range of α values, both positive and negative.

As can be seen in Figure 2-6, when constraints are relaxed, the range of costs increases dramatically. This is because when constraints are relaxed greatly, other infeasible solutions (as found by EPANET) are deemed feasible by the ANN. Therefore, in this case, the ANN-GA may quickly converge to an infeasible region and then continue searching in that local area. As a result, the only way the optimal solution can be found would be to conduct a local search around one of the first feasible solutions found by the ANN-GA.

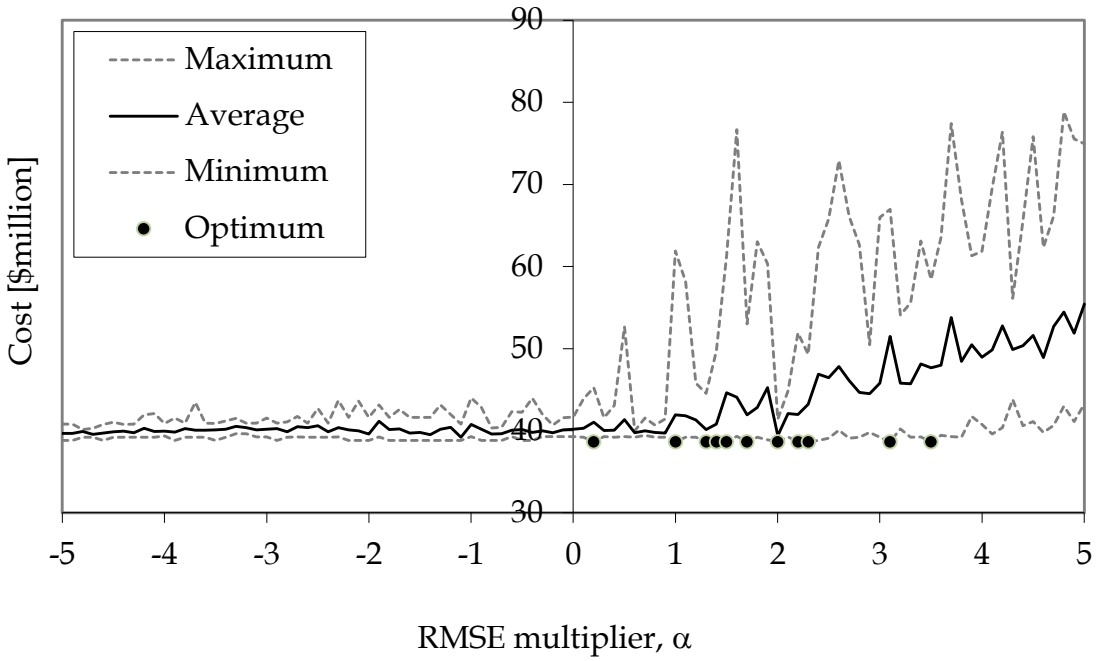


Figure 2-6. Constraint adjustment for the NYT-WQ problem:

ANN #3 of scenario A.

It is difficult, if not impossible, to determine by how much constraints should be relaxed (or tightened). At the extremes, if the constraints are relaxed too much, most of the search space would be deemed feasible by the ANN and the ANN-GA would converge to a solution that is infeasible when checked with EPANET. Conversely if constraints are tightened too much, the ANN would deem too little of the search space feasible and the ANN-GA would converge to a sub-optimal solution.

Other constraint adjustment results are given in Table 2.4. By adjusting constraints, all of the ANNs were now able to find the optimal solution. The range of α values for which the optimum is found is also given in Table 2.4, along with the solutions they found without constraint adjustment. An interesting feature is that ANNs that originally found good, but not-optimal solutions did not necessarily

perform better with constraints adjusted than ANNs that originally found poor solutions. For example, ANN #2 of scenario A found the optimum for a wide range of α values, whereas ANN #3 of both scenarios A and B only found the optimal solution for a narrow range of α. Two of the ANN metamodels in Table 2.4 found the optimal solution for either a relaxation of, or a tightening of constraints. This is probably due to the fact that separate ANNs were used to model each node and therefore one may be over-estimating pressure, while another under-estimates it.

Table 2.4. Optimal solutions found [$million] by constraint adjustment for the NYT-WQ problem.

Scenario	ANN Number	Solution Found (With Adjusted Constraints)	Main α Range For Finding Optimal Solution	Solution Found with α=0
A	1	38.64	[-1.5, 2.6]	38.64
	2	38.64	[-2.3, -1.1] & [0.6, 3.1]	39.06
	3	38.64	[1, 2.5]	38.80
B	1	38.64	[-0.3, -0.2] & [3.5, 4.8]	38.80
	2	38.64	[-4.7, -2.4]	38.80
	3	38.64	[2.8, 4.0]	39.21

2.7.6 Comparison of Computational Time

Table 2.5 gives a comparison of the time taken to optimize the NYT-WQ problem with EPANET-GA and the time needed to generate data, train the ANN and optimize the same problem with the ANN-GA. Also shown are the number of EPANET function calls during optimization.

Table 2.5. Computational times for metamodeling and optimization of the NYT-WQ problem.

Model	Sampling Data [hours]	Training ANN [hours]	Optimization [hours]	Total [hours]	EPANET Function Calls
EPANET	N/A	N/A	21	21	800,000
ANN	0.37	16	0.03	16.40	115

Note: the runtimes presented here are for a single optimization run with the metamodels using all three strategies to check solutions with EPANET.

The greatest computational burden in developing the metamodel comes from training. The training time for the ANNs was 16 hours. Combining this with optimization time and time to generate training data results in an overall time saving of 21% compared with the optimization time for EPANET-GA.

The time needed to optimize the problem once the metamodel is developed is only a fraction of the time needed to optimize with EPANET. Ignoring the computational time needed to develop the metamodel, the ANN-GA runs 700 times faster than the EPANET-GA.

2.7.7 Discussion

The NYT problem involves exceptionally large pipe diameters (up to 5 meters), atypical of those in other optimization problems in the literature. Large pipes correspond to small headlosses, hence different pipe sizes in a certain segment would result in little variation in pressure at the downstream node. As the ANNs are capable of adequately modeling such small variations in pressure and still able to obtain the optimal solution to the NYT-WQ problem, this metamodeling approach should apply well to other WDS optimization problems.

More complicated WDS may have 20 or 30 critical nodes. If separate ANNs were trained for each of these nodes, the computational time may be excessive, to the point where there might be no net benefit in using a metamodelling strategy. Therefore, the effect of other ANN architectures should be investigated further. For example, WDS problems with many critical nodes could be to train a few ANNs each with a number of outputs.

2.8 Other Applications for WDS

The vast difference between optimization time for the ANN-GA compared to EPANET-GA highlights the added value of using a metamodel for optimizing WDS operations. This is generally carried out on an hourly, daily or weekly schedule, for example. Metamodeling, therefore, has additional benefit for an operations problem over design in that the ANN only needs to be trained once, and then can be used repeatedly for optimization. An operations-based ANN would have different inputs than one used for design. In this case, training data would possibly consist of different values for tank operating levels, pump and valve settings, and the location and setting of chlorine booster stations. For this approach to be feasible, training data would need to be generated across the whole range of possible solutions. This would require foresight by the engineer of all possible solutions and thus is not a trivial matter. However, this does provide new possibilities for research in this field.

Another possible application, and one with greater potential in terms of time-saving is in evaluating system reliability. In a similar way to that described in this paper, ANNs could be trained to reduce the run-time required in a Monte Carlo simulation (MCS). One approach would be to train an ANN to approximate pressures, with additional inputs of demand and other stochastic variables. This ANN could then

be used in place of EPANET in a MCS. Another approach would be to train an ANN to approximate reliability directly, thus eliminating altogether the need to conduct a MCS. This is a particularly useful application, considering the amount of research being conducted in reliability-based optimization in recent years. The number of MCS realisations in a single reliability calculation may be in the order of 10,000 and the results in this paper indicate the ANN-GA runs 700 times faster than EPANET-GA. Hence reliability-based optimisation with ANNs could potentially run 7 x 10^6 times faster than if EPANET was used.

2.9 Conclusions

The results presented in this paper illustrate the validity of a new approach to WDS optimization. ANN metamodels linked to a GA were able to find the same optimal solution ($38.64m) as EPANET linked to a GA, for the NYT-WQ problem. Also, the total computational time was lower for optimization with an ANN-linked to a GA (16 hours, including generating data, training the ANNs and optimization) than EPANET-linked to a GA (21 hours). A comparison of optimization runtimes shows that a trained ANN runs approximately 700 times faster than EPANET for the NYT-WQ problem.

There are many factors that affect the ability of the ANN-GA to find the optimal solution. Initial weights in the ANNs influenced the accuracy of the final trained ANNs, which in turn influenced the quality of the optimal solution found by the ANN-GA. Also affecting the optimal solution was the GA itself. Given that EPANET-GA could only find the optimal solution on three out of five runs, it is encouraging that one of the trained ANNs found the same optimum on five out of ten runs. The randomly generated data did not appear to have a significant impact on the results. While

scenarios A and B had similar results to each other in terms of RMSE and little difference in terms of the optimal solutions they found, there may still be benefit in trying different methods for generating the training data, other than doing so randomly.

The metamodeling technique significantly reduces the high run-times, which are an unfortunate feature of WDS optimization using GAs. The technique is useful for significantly reducing optimization time on existing problems, but now also allows the focus of optimization to be expanded to include both reliability and water quality. It is unlikely that an ANN could be trained that is a perfect approximation to the specific EPANET model of the WDS. To combat this problem, two techniques have been presented here, including to evaluate select solutions with the original simulation model and to adjust constraints slightly so the optimal solution can be found. These two techniques alleviate the problem of imperfect metamodels and make optimization significantly faster than if the simulation model was used.

2.10 Acknowledgments

The authors would like to thank the Co-operative Research Centre for Water Quality and Treatment, based in Adelaide, Australia, and the Australian Department of Education, Science and Training for their financial support of this project. The authors would also like to thank the three reviewers for their comments.

Chapter 3

Publication 2: Local Search

"...seek and you will find..."

Matthew 7:7b (NIV)

Statement of Authorship

Title of Paper	Improving Metamodel-based Optimization of Water Distribution Systems with Local Search.
Publication Status	Published
Publication Details	Broad, D. R., Dandy, G. C., Maier, H. R., and Nixon, J. B. (2006). "Improving Metamodel-based Optimization of Water Distribution Systems with Local Search." *IEEE World Congress on Computational Intelligence*, 16-21 July 2006, Vancouver, BC, Canada.

Author Contributions

Name of Co-Author	Darren Broad		
Contribution to the Paper	Conceptual and theoretical development, interpretation and analysis of results, manuscript preparation and corresponding author.		
Signature		Date	

Name of Co-Author	Graeme Dandy		
Contribution to the Paper	Project supervision and review of manuscript.		
Signature		Date	

Name of Co-Author	Holger Maier		
Contribution to the Paper	Project supervision and review of manuscript.		
Signature		Date	

Name of Co-Author	John Nixon		
Contribution to the Paper	Project supervision and review of manuscript.		
Signature		Date	

Although the manuscript has been reformatted in accordance University guidelines, and sections have been renumbered for inclusion within this thesis, the paper is otherwise presented herein as published.

Abstract

Metamodels can be used to aid in improving the efficiency of computationally expensive optimization algorithms in a variety of applications, including water distribution system (WDS) design and operation. Genetic Algorithm (GA)-based optimization of WDSs is very computationally expensive to optimize a system in a practical amount of time for real-sized problems. A metamodel, of which Artificial Neural Networks (ANNs) are an example, is a model of a complex simulation model. It can be used in place of the simulation model where repeated use is necessary, such as when carrying out GA optimization. To complement the ANN-GA, six local search algorithms have been developed or applied in this research, with the aim of improving the performance of metamodel-based optimization of WDSs. All algorithms performed well, however, using computational intensity as a criterion with which to evaluate results, the best local search algorithms were Sequential Downward Mutation (SDM) and Maximum Savings Downward Mutation (MSDM).

3.1 Introduction

Water distribution systems (WDSs) are complex systems whose optimal configuration is best determined by Evolutionary Computation (EC) techniques, such as Genetic Algorithms (GAs) (Simpson et al. 1994). There are many examples in the literature that illustrate the applicability of GAs to WDSs (Simpson et al. 1994; Walters et al. 1999; Halhal et al. 1997; Dandy et al. 1996; Savic and Walters 1997; Wu and Simpson 2001). These examples showed improvement over earlier optimization techniques used in the field, including linear programming (Morgan and Goulter 1985), enumeration (Gessler 1985; Walski et al. 1987) and gradient techniques (Walski et al. 1987; Duan et al. 1990; Lansey and Mays 1989; Ormsbee and Kessler 1990; Sakaraya and Mays 2000; Xu and Goulter 1999).

WDS optimization can be classified into one of two categories; design or operation. WDS design involves determining the optimal combination of new infrastructure (e.g. pipes, pumps, tanks) that is to be constructed, and hence capital costs are important. Conversely, WDS operation involves determining the optimal set of settings for existing infrastructure (e.g. chlorine dosing rates, tank operating levels), and hence on-going costs, such as electrical, chemical and maintenance costs, are of prime concern. Therefore, in general a WDS configuration, Ω, consists of a set of n decisions (Zecchin et al. 2005), as given by Eq. (3.1).

$$\Omega = \{decision_1, \ldots, decision_n\} \tag{3.1}$$

For each decision, there are a number of options, as given by Eq. (3.2).

$$decision_i \in \{option_{i,1}, \ldots, option_{i,NO_i}\}, \ \forall\, i = 1,\ldots,n \tag{3.2}$$

Where $option_{i,j}$ is the *j*-th option for decision *i*, and NO_i is the number of possible options for decision *i*. Hence, the total number of possible configurations for a WDS is given by Eq. (3.3).

$$N_{configurations} = \prod_{i=1}^{n} NO_i \tag{3.3}$$

This value can be in the order of 10^{25} for moderately sized systems (Dandy et al. 1996) and up to 10^{74} for larger systems (Halhal et al. 1997). This has led to the use of Genetic Algorithms (GAs) as a technique that can obtain optimal or near-optimal solutions within a practical amount of time (Simpson et al. 1996; Dandy et al. 1996; Savic and Walters 1997). Further research in the field of WDS optimization focused on the incorporation of water quality constraints into the problem formulation to complement hydraulic constraints and thus result in more robust WDS configurations (Dandy and Hewitson 2000).

A generalized formulation for the WDS optimization problem is given by Eqs. (3.4-3.6). The objective function is to minimize some cost function by selecting the best set of decision variables such that certain hydraulic constraints (e.g. maximum and minimum pressure heads, pipe velocities) and water quality constraints (e.g. maximum and minimum residual chlorine, or particle concentrations) are met.

$$\min z = f(\Omega) \tag{3.4}$$

Such that

$$\underline{H_i^j} \leq H_i^j(\Omega) \leq \overline{H_i^j}\ ,\ \forall\, i\ =\ 1, \dots, \mathrm{N},\ \forall\, j\ =\ 1, \dots, T \tag{3.5}$$

$$\underline{C_i^j} \leq C_i^j(\Omega) \leq \overline{C_i^j}\ ,\ \forall\, i\ =\ 1, \dots, \mathrm{N},\ \forall\, j\ =\ 1, \dots, T \tag{3.6}$$

Where $\underline{H}$ and $\overline{H}$ are the minimum and maximum hydraulic constraints; $H(\Omega)$ is the actual hydraulic performance value for configuration Ω; $\underline{C}$ and $\overline{C}$ are the minimum and maximum water quality constraints; $C(\Omega)$ is the actual water quality performance value for configuration Ω; N is the number of nodes or pipes in the WDS (depending on the specific type of constraint); T is the time horizon for the optimization (approximately 24-48 hours). The detail of the objective function in (4) depends on whether the problem is for the design case or for operations and is also specific to each WDS. The reader is referred to Zecchin et al. (2005) for further detail of the hydraulics of WDSs and Boccelli et al. (1998) for further detail of the kinetics of chlorine in WDSs. Constraint Eqs. (3.5) and (3.6) are typically accounted for through the use of penalty costs, hence the objective function that can be used in a GA is given by Eq. (3.7).

$$\begin{aligned} \min z = f(\Omega) &+ \max_{i,j}\left[\max\left\{0, \left(\underline{H_i^j} - H_i^j(\Omega)\right), \left(H_i^j(\Omega) - \overline{H_i^j}\right)\right\}\right] PM_1 \\ &+ \max_{i,j}\left[\max\left\{0, \left(\underline{C_i^j} - C_i^j(\Omega)\right), \left(C_i^j(\Omega) - \overline{C_i^j}\right)\right\}\right] PM_2 \end{aligned} \tag{3.7}$$

Where PM_1 and PM_2 are the respective penalty multipliers for hydraulics and water quality.

Unfortunately this formulation results in very long runtimes, as hydraulic computer simulation models with which a given configuration is evaluated, such as EPANET (developed by the US EPA), are much more computationally intensive when a water quality simulation is required. This issue of computational intensity has led to the use of metamodels for WDS optimization (Broad et al. 2005a). A metamodel is a model of a complex simulation model (Blanning 1975) that can be used in place of the simulation model where repeated use is necessary, such as in EC-based optimization.

Metamodels approximate the input/output transformation that is implied by the simulation model. In general, a metamodel maps the outputs as a function of the inputs, resulting in what is known as a response surface (Kleijnen and Sargent 2000).

Many metamodel examples (Rogers and Dowla 1994; Aly and Peralta 1999; Johnson and Rogers 2000; Neelakantan and Pundarikanthan 2000) directly approximate the objective function, however, in the case of WDS optimization this is impractical and unnecessary as (i) the penalty costs result in an objective function that has discontinuities and is therefore difficult to approximate with a metamodel and (ii) the computational burden arises from calculating the constrained variables and not in calculating penalty and actual costs. Hence metamodels for WDSs generally approximate constrained variables (outputs) as a function of decision variables (inputs) (Broad et al. 2005).

Artificial Neural Networks (ANNs) have been used previously as metamodels for various simulation models that have high computational cost (Broad et al. 2005; Rogers and Dowla 1994; Aly and Peralta 1999; Johnson and Rogers 2000; Neelakantan and Pundarikanthan 2000; Lingireddy and Ormsbee 1998) ANNs (specifically, Multi-Layer Perceptrons) were chosen as the metamodel-type in this research due to their proven performance and their ability to approximate any continuous function (Leshno et al. 1993). This property is particularly appealing given that the equations that govern both the hydraulics and water quality in WDSs are non-linear.

While the theoretical capabilities of ANNs are valuable, in practice, it is difficult to construct ANN metamodels that are perfect representations of simulation models (Broad et al. 2005). However, in previous applications of ANNs for the

purpose of metamodelling, it was assumed that the metamodel was sufficiently accurate and the solution to which the ANN-linked GA converged was assumed to be the actual optimum (Rogers and Dowla 1994; Aly and Peralta 1999; Johnson and Rogers 2000; Neelakantan and Pundarikanthan 2000; Lingireddy and Ormsbee 1998). However, this is unlikely to be the case, as even small errors in the metamodel can result in the acceptance of infeasible solutions or the rejection of feasible solutions as part of the optimization process. In order to overcome this shortcoming, (Broad et al. 2005) introduced a three-stage method of checking solutions, including (i) checking each new best solution with the simulation model, (ii) tracking the best 40 solutions which are evaluated using the simulation model, and (iii) conducting a local search using the simulation model. This paper further explores the effectiveness of that third stage by considering a range of different local search algorithms.

3.2 Metamodelling Procedure

The processes that need to be followed in developing an ANN metamodel to be used in place of a simulation model for optimization are shown in Figure 3-1. The first step is to set up the problem that is to be optimized, including defining the objective function, decision variables and constraints. Next, training data need to be generated with the simulation model. Broad et al. (2005) found that as few as 10,000 solutions are needed to construct an adequately accurate metamodel for the purpose of WDS optimization. This is likely to be problem dependent, however the same amount of data was used in this research. Data are generated randomly in the search space across the whole range for each of the decision variables. The reason for this is that in practice, one would not know *a priori* where in the search space the optimum is located.

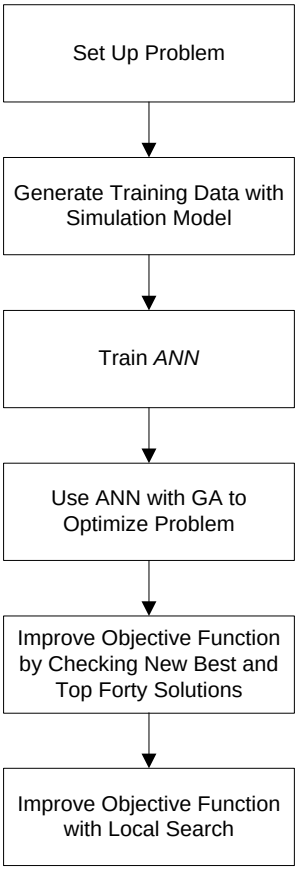


Figure 3-1. Procedure for developing and using an ANN metamodel for the purpose of optimization.

Given that an ANN metamodel is trained using synthetic data generated from a simulation model, there will not be any noise in the data (Johnson and Rogers 2000). This is not usually the case in many real world applications of ANNs. Hence, rather than minimizing the error in a test set as is generally the case when noise is present, a fixed number of iterations should be used. Otherwise the error continually decreases, without over-training, even with a large number of iterations. The number of iterations required for sufficient accuracy needs to be determined from a series of preliminary runs.

The next step is to optimize the problem with the GA, using the trained ANN in place of the simulation model. Each new best solution the ANN-GA finds must be checked with the simulation model for feasibility. This is a necessary step, as one cannot simply assume that the ANN will be a perfect representation of the simulation model (Broad et al. 2005), as discussed above.

The final step is to perform a local search with a satisfactory algorithm to improve on the solution obtained by the ANN-GA. This paper focuses on that final step of the metamodelling optimization process. Broad et al. (2005) and Lingireddy and Ormsbee (1998) provide greater detail on how to construct metamodels for the purpose of WDS optimization.

3.3 Local Searches

As mentioned in the introduction, local search is a necessary final step in the ANN-GA process. As the local search is complementary to the ANN-GA search, algorithms were selected based on certain criteria, including (i) simplicity of implementation (derivatives were to be avoided as the evaluation of the objective function required an external solver (EPANET), hence gradient techniques were not considered); (ii) it was assumed the ANN-GA would find near-optimal solutions, hence the local search would concentrate on the surrounding neighborhood of the best solution found by the ANN-GA.

In this paper, six local search algorithms were either developed or applied to metamodel-based optimization of WDSs, which are detailed in the subsequent sections.

3.3.1 Sequential Downward Mutation

This algorithm was devised to obtain improvements in the solution after the ANN-GA had executed. It was named 'Downward Mutation' because it was originally applied to a pipe network optimization problem, where a lower cost results from a decrease in pipe size for all alleles in the string. The pseudo-code for Sequential Downward Mutation (SDM) is shown in Figure 3-2; where A is the name of the solution string; and f( ) is the objective function. 'Sequential' refers to the order in which the alleles are selected. The first allele in the string is selected as the starting point and the value of that allele is reduced incrementally until there is no further improvement, at which time the search moves onto the second allele. Hence, each allele, or decision variable, is searched sequentially for improvements in the objective function. 'Downward' refers to the direction along which the objective function is reduced locally. Therefore, this technique may not be easily applied to other types of optimization problems. For example, if alleles in the string represent tank operating levels, it is not clear whether reducing the level will result in an overall reduction in cost. In addition, the specification of smaller pipes results in lower capital cost, and the use of a lower chlorine dosing rate results in lower operating costs.

```
set i = 1
do
      set A'[i] = A[i] - 1
      if f(A') < f(A)
            set A = A'
      else
            set i = i + 1
            if i = dim (A) + 1
                  stop
loop
```

Figure 3-2. Pseudo-code for sequential downward mutation

3.3.2 Random Downward Mutation

A variation on SDM is Random Downward Mutation (RDM), wherein the order in which alleles are selected for downward mutation is randomized. Rather than mutating down each allele in a sequential order, an allele is selected at random and its value is reduced by one. If this results in a better value of the objective function, the string is updated. At the second, and subsequent iterations, alleles are again selected at random and the process continues until no further improvement can be obtained. The pseudo-code for Random Downward Mutation (RDM) is shown in Figure 3-3.

```
do
      select i randomly ∈ [1,dim (A)]
      set A'[i] = A[i] – 1
      if f(A') < f(A)
            set A = A'
            count = 0
      else
            count = count + 1
            if count = dim (A) + 1
                  stop
loop
```

Figure 3-3. Pseudo-code for random downward mutation

3.3.3 Maximum Savings Downward Mutation

Another variation of the previous two downward mutation algorithms is to use a more intelligent method of selecting in which direction to search, which is termed the Maximum Savings Downward Mutation (MSDM) algorithm. The pseudocode for this algorithm is shown in Figure 3-4, where g( ) is the actual cost (i.e. the objective function minus penalty costs for constraint violation). The search works by iteratively selecting the allele in the string that will give the greatest saving, or

reduction in the objective function. For example, in a design situation, where each allele represents the diameter of a pipe segment in a WDS, reducing the pipe size of the longest pipe segment would generally result in the greatest savings. It was envisaged that by selecting to mutate the allele that would provide the greatest savings, the local search would converge faster than either selecting the allele sequentially (i.e. SDM) or randomly (i.e. RDM).

```
do
      select i: max(g(A[i]-1) – g(A[i]))
      set A'[i] = A[i] - 1
      if f(A') < f(A)
            set A = A'
            count = 0
      else
            count = count + 1
            if count = dim (A) + 1
                  stop
loop
```

Figure 3-4. Pseudo-code for maximum-savings downward mutation

3.3.4 Triangular Mutation

In an attempt to avoid becoming trapped in any local optima, which are common in WDS optimization problems (Gibbs et al. 2004), a local search named Triangular Mutation (TM) was devised. It is an adaptation of Adjacency Mutation (Dandy et al. 1996), which differs from ordinary mutation in that the allele can only mutate to values adjacent to the current value. Ordinary mutation, however, allows the value of the allele to mutate to any other value. TM extends the concept of adjacency mutation by allowing the value of the allele to mutate more than one unit, but with a decreasing probability. The probability density function is triangular in

shape on both sides of the selected allele, A[i], hence the name triangular mutation. Each of these types of mutation is illustrated in Figure 3-5.

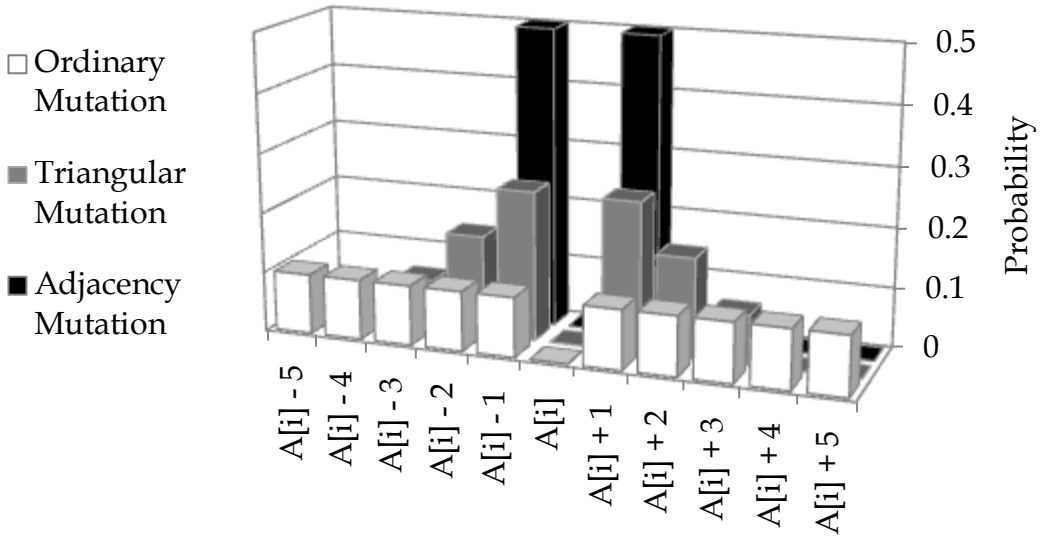


Figure 3-5. Comparison of different types of mutation.

The pseudocode for TM is shown in Figure 3-6, where Pm* is the probability of mutating an allele in the string; Pr(x) is the probability of mutating to x; γ is the maximum number of units to which an allele can be mutated; R( ) is the biased roulette wheel function and randomly generates a value for A′[i] based on the probabilities calculated.

The other main feature of triangular mutation is that it requires a higher probability of mutation than that which is used in the GA. This is needed because mutation is the only means by which this local search operates, rather than being an operator in a GA. A higher probability of mutation ensures that at least one allele in the string is mutated, which in turn, ensures there are no iterations with solutions exactly the same as in the previous iteration.

```
do
      for i = 1 to dim (A)
            set a = U[0,1]
            if (a <= Pm*)
                  set Pr(A[i] + 1) = 1/(γ + 1)
                  set Pr(A[i] - 1) = 1/(γ + 1)
                  for j = 2 to β
                        set b = (γ − j + 1)/(γ − j + 2)
                        set Pr(A[i]+j)=b.Pr(A[i]+(j-1))
                        set Pr(A[i]-j)=b.Pr(A[i]-(j-1))
                  next j
                  set A'[i] = R(A[i])
      next i
      if f(A') < f(A)
            set A = A'
      count = count + 1
      if count = max_iterations
            stop
loop
```

Figure 3-6. Pseudo-code for triangular mutation

3.3.5 Probabilistic Allele Swapping

The Probabilistic Allele Swapping (PAS) local search is an adaptation of a local search that was specifically devised for the traveling salesman problem, which was called Compounded Swaps (Ahuja et al. 2002). The objective of the traveling salesman problem is to minimize the distance a salesman would need to travel, while visiting a certain group of cities and visiting each only once. The solution string represents the order in which the cities are visited. A Compound Swap involves randomly selecting two cities and swapping them in the string.

The compound swaps method can be used directly for WDS optimization. Rather than swapping the order in which cities are visited, the diameters of two pipes could be swapped. However, this would result in each new string being far different than in the previous iteration. It is assumed that the starting position in each of these local searches is not far from the global optimum, hence each iteration of a local search should only involve small differences between the old and new strings.

```
do
      select i randomly ∈ [1,dim (A)]
      select j randomly ∈ [1,dim (A)]
      set A'[i] = A[i] - 1
      set x = U[0,1]
      if (x < ρ)
            set A'[j] = A[j] + 1
      if f(A') < f(A)
            set A = A'
      count = count + 1
      if count = max_iterations
            stop
loop
```

Figure 3-7. Pseudo-code for probabilistic allele swapping

Therefore, the compounded swaps local search has been adapted, the pseudocode of which is shown in Figure 3-7. Two alleles are randomly selected. Then, rather than swapping the diameters of those pipes, one is increased slightly and one is decreased slightly. Now, if two pipe diameters are increased and decreased, respectively, only limited improvements in the objective function can be achieved. Therefore, another aspect of this local search is that the allele that is selected to increase, is only increased with a given probability, ρ. Hence the name given to this

local search is Probabilistic Allele Swapping, as it is derived from the Compounded Swaps algorithm.

This local search is particularly suited to WDS optimization. Each decision variable in the string represents a single pipe in the network. Therefore, if one pipe segment is incremented up to the next diameter, while one is reduced to a lower diameter, it is likely that the net result will only be a slight difference in flow paths through the system and pressures at the critical nodes will only be changed slightly.

3.3.6 Simulated Annealing

The local search methods detailed previously were either developed as completely new algorithms or adapted significantly from existing techniques. Simulated Annealing (SA), however, is an existing method, that has also been applied to WDS optimization (Cunha and Sousa 1999). In that case, SA was used as the sole optimization technique, whereas here it is proposed to be used after the ANN-GA has converged and therefore acts as a fine-tuning technique to slightly reduce the value of the objective function.

As the name suggests, simulated annealing is analogous to the annealing process used in developing the crystalline structure in metals. Initially, the temperature of the material is high and the crystalline structures are less stable. As the temperature decreases, the crystals become more rigid and stable. In simulated annealing, initially, solutions to the problem that are worse than in the previous iteration are accepted with a relatively high probability. As the algorithm progresses, the 'temperature' decreases, and worse solutions are accepted with lower probability. During the physical annealing process, the temperature must be controlled so as to not allow warping or cracking in the material. So too with SA: the 'temperature' must

be controlled so as to avoid premature convergence of the algorithm to a sub-optimal solution.

While TM and PAS have properties that enable the search to escape local optima, SA actually accepts worse solutions during the search, thus enabling the search to 'climb over' peaks in the search-space; whereas the manner in which TM and PAS escape local optima is by using larger steps in the search process.

```
do
      select i randomly ∈ [1,dim (A)]
      set x = U[0,1]
      if x < 0.5
            set A'[i] = A[i] - 1
      else
            set A'[i] = A[i] + 1
      set y = U[0,1]
      set z = min(1, exp((f(A') - f(A))/t))
      if y < z
            set A[i] = A'[i]
      t = r.t
      if t < t_min
            stop
loop
```

Figure 3-8. Pseudo-code for simulated annealing.

The pseudocode for SA is shown in Figure 3-8. At each iteration, a new solution is generated by increasing or decreasing one randomly selected allele by one unit. If this new solution is better than the previous, it is accepted. However, if the solution is inferior, it still may be selected, but with a probability, *z*. The temperature, *t*, decreases with each iteration, hence z also decreases, as the search progresses.

3.4 Case Study

The New York Tunnels (NYT) problem (Schaake and Lai 1969) was chosen as the case study on the basis that there has been considerable research conducted on it in the past and therefore the current best solution is probably close to, if not *the*, global optimum. Therefore, this enables the effectiveness of the local search methods to be evaluated. The NYT problem is a WDS expansion problem, where the optimal set of diameters of pipe segments need to be determined such that pressure at all nodes is above a specified minimum for a given set of demands. Further details of the NYT problem can be found in Maier et al. (2003), which also contains the current best known solution of $38.64M when EPANET 2.0 is used as the hydraulic solver.

Broad et al. (2005) created the NYT water quality (NYT-WQ) problem by adapting the NYT problem to include water quality constraints in the form of chlorine residual concentrations in the system. The NYT-WQ problem consists of 22 decision variables, including 21 pipe diameters and one chlorine dosing value. There are 16 possible diameters for each pipe in the system and 21 possible chlorine dosing concentrations, resulting in a search space of 4.1×10^{26} possible solutions.

It was found that approximately 10,000 data were needed to construct adequately accurate metamodels, which was determined from several preliminary runs. From the training data that were generated, it was found that 4 of the junctions in the system were critical, meaning that the minimum pressure in the system can occur at any one of four junctions. Similarly, there is one critical junction for the chlorine residual constraints. These critical junctions, along with a layout of the NYT system, are shown in Figure 3-9.

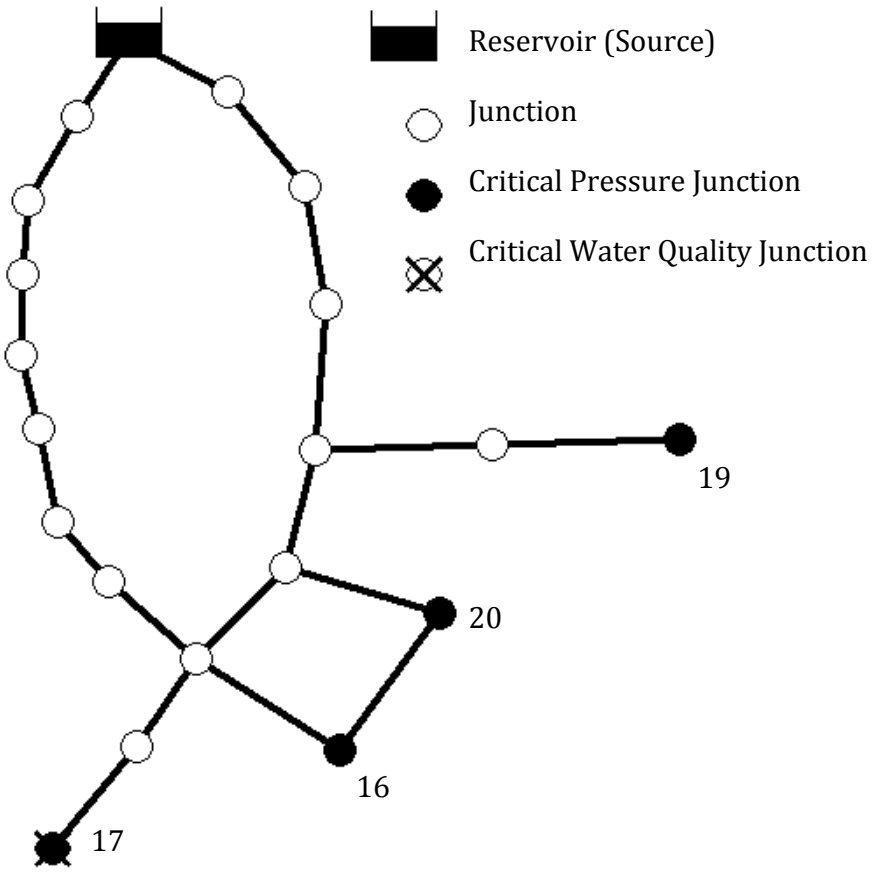


Figure 3-9. The New York Tunnels Water Distribution System, with critical nodes.

3.4.1 Analysis Conducted

To enable valid comparisons to previous work, the same ANN and GA parameters were used as in Broad et al. (2005). These values, shown in Tables 3.1 and 3.2, were calibrated to achieve fast convergence and maximum performance (i.e. small error for the ANNs and near-optimal objective function for the GA). Note the penalty multipliers for pressure head and chlorine residual are measured in $/m and $/mg/L, respectively.

As listed in Table 3.1. ANN Parameters, a multi-layer perceptron was used with 40 hidden nodes. As there are 22 decision variables, the number of input nodes

for the metamodel is also 22. There are a total of 5 critical junctions (4 for pressure and 1 for chlorine), hence there are 5 output nodes for the metamodel. The metamodel consisted of 5 separate ANNs, each with a single output, rather than a single ANN with 5 outputs. Preliminary results showed this was the more accurate approach, but for brevity these results are not presented here. The hydraulic WDS simulation model EPANET 2.0 was run 10,000 times with different randomly selected values for the 22 decision variables and five corresponding metamodel outputs, to obtain the training data.

To examine the effect the initial weights in the ANNs have on the quality of solutions obtained, 30 different metamodels were trained, each with different randomly generated initial weights. Similarly, to reduce the effect of random initialization of the GA population, each optimization run was carried out 100 times. The optimization was carried out with a metamodel that was representative of the 30 that were trained. As there was very little variance in the accuracy of the metamodels, this was considered to be a valid approach.

Table 3.1. ANN Parameters

Parameter	Value
ANN Type	multi-layer perceptron
Transfer Function	sigmoid
No. Hidden Layers	1
No. Hidden Nodes	40
Learning Rate	0.3
Momentum Rate	0.5
Training Iterations per ANN	5000

Table 3.2. GA Parameters

Parameter	Value
GA Type	integer
Population Size	400
Probability of Crossover	0.8
No. Crossovers per Pair	1
Probability of Bit-Wise Mutation	0.02
No. Generations	2000
Penalty Multiplier – Pressure Head	10^9
Penalty Multiplier – Chlorine Residual	10^9

The three downward mutation algorithms were run until no further improvement was possible in each of the alleles of the string. By their nature, there is a finite number of iterations for these algorithms. Conversely, each of the other local search algorithms were run for as many iterations as was required for convergence, where convergence was defined as no improvement in the objective function in the preceding 50 iterations. Also, for algorithms that involve parameters, the best values of the parameters were calibrated using a simple sensitivity analysis, the outcomes of which are shown in Table 3.3. As expected, the probability of mutation for TM is higher (0.1) than that which was used for the GA (0.02). The maximum mutation distance for TM was 3, but this is likely to be problem specific. A value of 0.5 was selected for the probability of increasing an allele, but this parameter was actually fairly insensitive. The best cooling rate for SA was found to be 0.8. Higher values resulted in premature convergence to sub-optimal solutions, while lower values resulted in the algorithm continually searching without converging.

Table 3.3. Calibrated local search parameters.

Local Search	Parameter	Parameter Symbol	Value
TM	Prob. Bit-wise Mutation	Pm*	0.1
TM	Max. Mutation Distance	γ	3
PAS	Prob. Increasing Allele	ρ	0.5
SA	Cooling Rate	r	0.8

3.4.2 Results

The accuracy of the metamodels is important in determining the effectiveness of the ANN-GA. The RMS errors in the validation set from the 30 metamodels that were developed are presented in Table 3.4, where pressure heads (H) are measured in [m] and chlorine residuals (C) are measured in [mg/L]. These errors are quite low, especially in comparison to the range covered by the training data, which is also presented in Table 3.4. The errors were approximately 0.5% of the range for each ANN output.

Table 3.4. RMS error in the validation set and training data range from 30 ANN Metamodels

ANN Output	RMS error		Range
	Average	St. Dev.	
H (node 16) [m]	0.124	0.007	64.5 – 89.5
H (node 17) [m]	0.027	0.003	80.9 – 89.5
H (node 19) [m]	0.254	0.032	30.1 – 89.4
H (node 20) [m]	0.125	0.007	64.1 – 89.5
C (node 17) [mg/L]	0.0053	0.0001	0.004 – 0.745

After the ANN metamodels have been trained, the next stage is the optimization with the ANN-GA, followed by local search. The results from each of the local search algorithms are presented in Table 3.5, as well as the results when no local search was utilized. The minima, average and maxima of the 100 optimization runs are presented for each algorithm. It can be seen that each algorithm obtained the same optimum as Broad et al. (2005) at least once, whereas there is some variance in the averages. It can be seen that the maximum is much greater than the average. This was due to the fact that there was one GA run that converged to a particularly poor solution and so the local search was not capable of providing significant improvement in the objective function due to a poor starting position.

To examine the significance of the local search results presented in Table 3.5, a t-test was performed, with the results given in Table 3.6. At the 95% significance level, it is clear that all the local search algorithms performed better than when no local search was used. The difference between the local searches, however, was less noticeable. There was only a significant difference in the results between two pairs. SDM was significantly better than RDM, which was surprising, given that they are very similar algorithms. Also, SDM was significantly better than TM. These results indicate that while a local search is required to improve results, it is not important which algorithm is used.

Table 3.5. Local Search results from 100 random initialisations in the GA population.

Local Search	Min	Avg	Max
SDM	38.64	39.68	43.48
RDM	38.64	39.98	45.88
MSDM	38.64	39.89	43.67
TM	38.64	40.02	45.24
PAS	38.64	39.75	44.61
SA	38.64	39.93	44.86
None	41.60	42.41	48.13

Table 3.6. P values from a t-test, illustrating the significance of the local search results.

	SDM	RDM	MSDM	TM	PAS	SA	None
SDM		**0.025**	0.065	**0.012**	0.329	0.052	**2.8E-42**
RDM			0.284	0.406	0.069	0.379	**1.3E-33**
MSDM				0.202	0.158	0.406	**4.1E-38**
TM					0.292	0.292	**2.1E-33**
PAS						0.124	**3.9E-39**
SA							**3.6E-34**
None							

The quality of the solutions obtained is important in evaluating the performance of the local search algorithms. However it is also important to consider the computational intensity, particularly given that the purpose of metamodelling is to reduce the runtimes to a reasonable level such that WDS optimization can be achieved in practice. Therefore, the runtimes for each of the local searches is shown in Figure 3-10, as well as the runtime for ANN-GA without any local search and the runtime for EPANET-GA.

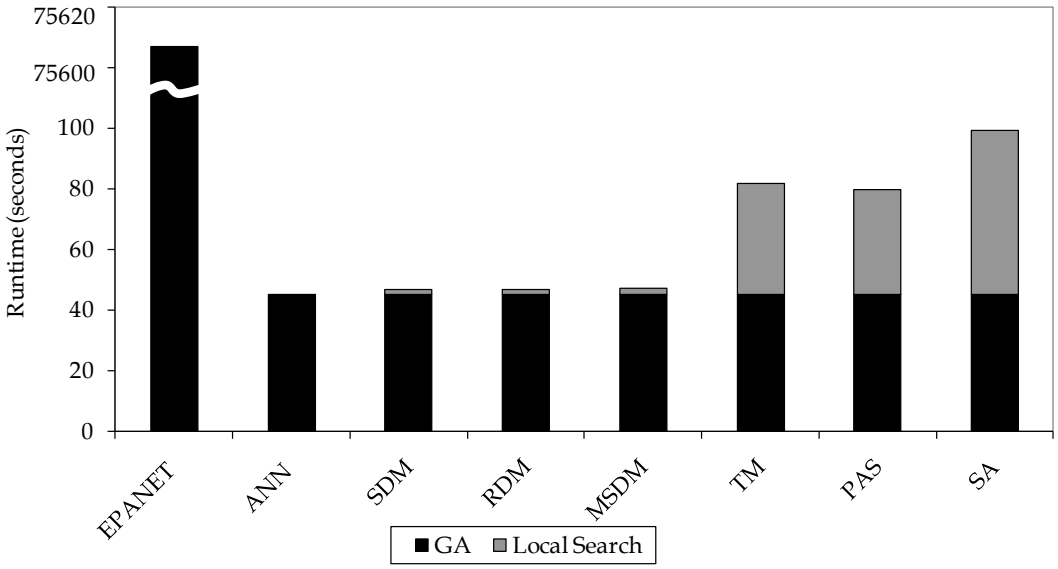


Figure 3-10. Run-times for each local search, with comparison to EPANET-GA and ANN-GA.

It can be seen that the ANN-GA takes approximately 45 seconds to run without any local search. The three downward mutation algorithms (SDM, RDM and MSDM) require negligible additional computational effort, adding only a few more seconds to the total time. The other three local searches (TM, PAS and SA) are all much more computationally intensive, with SA taking the most time to run at around 100 seconds. However, even though this is more than twice the runtime of ANN-GA without local search, it is still extremely short compared with the 21 hours that would have been required had ANN metamodels not been employed and EPANET used instead.

3.5 Conclusion

Six local search algorithms have been presented and evaluated in this paper for the purpose of improving the performance of metamodel-based optimization of

water distribution systems. With the exception of Simulated Annealing, they were developed for this specific purpose in mind.

The results show a significant improvement in the value of the objective function by using a local search as a complementary stage of metamodel-based optimization of WDSs. Closer examination of the local search results showed that it was not important which algorithm was used, as all the local searches considered here performed well. Hence, to determine which is the best type of local search, the respective runtimes were considered. The downward mutation algorithms were all quite fast, increasing the runtime of the ANN-GA only slightly. This runtime was much lower than the other three algorithms, hence considering both performance and computational requirements, either SDM or MSDM should be used.

The results presented in this paper were specifically developed for WDS optimization. However, they could all easily be applied to other metamodel applications with very minor modifications. The conclusions made, based on the results, are for one case study only. Further research will be conducted through CRCWQT Project 2.5.0.3 to determine whether the performance of the local search is dependent upon the shape of the fitness function, and hence performance may also depend upon the problem that is being optimized. This will involve application of the aforementioned techniques to further case studies from the literature, as well as a real WDS (Wallan) situated in Melbourne, Australia.

Chapter 4

Publication 3: Complex Hydraulic Systems

"When the Queen of Sheba heard about the fame of Solomon and his relation to the name of the LORD, she came to test him with hard questions."

1 Kings 10:1 (NIV)

Statement of Authorship

Title of Paper	Optimal Operation of Complex Water Distribution Systems Using Metamodels
Publication Status	Published
Publication Details	Broad, D. R., Maier, H. R., and Dandy, G. C. (2010). "Optimal Operation of Complex Water Distribution Systems Using Metamodels." *Journal of Water Resources Planning and Management - ASCE*, 136(4), 433-443.

Author Contributions

Name of Co-Author	Darren Broad		
Contribution to the Paper	Conceptual and theoretical development, interpretation and analysis of results, manuscript preparation and corresponding author.		
Signature		Date	

Name of Co-Author	Holger Maier		
Contribution to the Paper	Project supervision and review of manuscript.		
Signature		Date	

Name of Co-Author	Graeme Dandy		
Contribution to the Paper	Project supervision and review of manuscript.		
Signature		Date	

Although the manuscript has been reformatted in accordance University guidelines, and sections have been renumbered for inclusion within this thesis, the paper is otherwise presented herein as published.

Abstract

Optimization of large and hydraulically complex water distribution systems (WDSs) is computationally expensive as simulation models are required to evaluate the performance of solutions to the problem at hand. Metamodels can act as a surrogate or substitute for these simulation models and provide significant speed-ups in the optimization process. The application of metamodels in the field of WDS optimization has been limited to date, and little guidance has been given in terms of constructing metamodels for hydraulically complex systems. While it is relatively straightforward to obtain satisfactory metamodel approximations to simulation models of simple WDSs, this is not necessarily the case for more complex networks. In order to reduce the complexity of the relationship that is to be approximated by the metamodels, a number of factors have to be considered, including the complexity of the hydraulic simulation model, the complexity of the decision space, and the locations at which outputs are required from the hydraulic simulation model. This research presents a systematic methodology for dealing with these factors and demonstrates the effectiveness of the approach by applying it to an actual WDS.

A system in Wallan, Victoria, Australia, is selected for demonstration purposes. Four different metamodelling scenarios are presented here. The results show that, for this case study, some skeletonization of the model is required to achieve suitably accurate metamodels. The optimization results show a reduction in the average daily pumping costs from \$457 to \$363; a saving of 21%. The net present value (NPV) over 25 years is used as the objective function, which includes both pumping and chlorine costs. The current operating regime corresponds to an NPV of \$1.56 million, while the optimized solution has an NPV of \$1.34 million; a saving of 14%. In addition to these economic

benefits, the optimized solution achieves adequate disinfection throughout the system, whereas the current operating regime corresponds to deficits in chlorine residuals at several locations in the system.

4.1 Introduction

Determining the optimal operating strategy of Water Distribution Systems (WDSs) requires consideration of several factors, including the minimization of energy and disinfection costs, while satisfying customer requirements, such as providing adequate pressure and water quality. Energy costs are typically reduced by pumping during off-peak electricity tariff times as much as possible without compromising hydraulic performance (Mackle et al. 1995, Van Zyl et al. 2004), while disinfection costs are reduced by minimizing water age in the system (Boccelli et al. 1998). Consideration of water quality significantly increases computational intensity when modeling systems in a simulation package such as EPANET. Genetic Algorithms (GAs) have been shown to identify near globally optimal solutions to WDS optimization problems (Simpson et al., 1994), but their use becomes infeasible for realistic sized problems because of the long run times associated with the many hydraulic and water quality simulations. Consequently, there is a need to increase the computational efficiency of the WDS simulation model in order to address this problem. One way to achieve this is via network simplification approaches, such as skeletonization (e.g. Haestad Methods, 2002), decomposition (e.g. Deuerlein, 2008) or Gaussian elimination (e.g. Ulanicki et al., 1996). However, this is unlikely to achieve the several-orders-of-magnitude reduction in run-times needed. An alternative approach to increasing the computational efficiency of the simulation model is to use metamodeling (Blanning, 1975).

A metamodel is a surrogate, or substitute, for a computationally expensive simulation model, such as EPANET. In the context of GA optimization of WDSs, the simulation model is used to assess the fitness of a solution that is generated by the GA.

Consequently, the purpose of the metamodel is not to approximate the entire simulation model, but to obtain a relationship between the decision variables (e.g. chlorine dosing rates) and the constrained variables (e.g. chlorine residuals) and any other variables that contribute to the fitness (e.g. energy consumption). As this relationship is likely to be highly non-linear, Artificial Neural Networks (ANNs) have been used successfully for this purpose in various areas of water engineering (see Broad et al., 2005).

In relation to WDS optimization, Broad et al. (2005) introduced an approach for coupling a GA with an ANN metamodel and applied it to a modified version of the New York Tunnels WDS optimization problem (Schaake and Lai, 1969). Since then, there have been a number of similar studies, in which an ANN-GA metamodelling approach has been applied to the Anytown (Rao and Salomons, 2007), Haifa (Salomons et al., 2007) and Valencia (Martinez et al., 2007) WDSs.

When using metamodels in conjunction with GAs, a number of potential problems can arise. One of these is that even a small error in the metamodel can have a significant impact on the optimization results obtained. This is because metamodels are often used to check whether constraints (e.g. pressures, residual chlorine levels) have been satisfied. Consequently, even small errors in the metamodel can result in the inclusion of infeasible, or the exclusion of feasible, solutions, which can potentially lead the search to sub-optimal regions of the solution space.

This was recognized by Broad et al. (2005), who introduced a methodology for dealing with this issue, which includes the tracking of the best solutions (according to the metamodel) during the search, followed by their evaluation with the simulation model and a local search to fine-tune the solution. Broad et al. (2005) also proposed a

technique for adjusting constraints so that the approximated fitness landscape of the metamodel more closely matches the true fitness landscape. The approach was tested on a simple case study and was found to perform favorably in terms of accuracy and computational efficiency when compared with an optimization approach that combines EPANET with a GA (i.e. the same optimal solution was found at much reduced computational expense).

When dealing with realistic, rather than hypothetical, case studies, the problem of ANN accuracy is likely to be exacerbated due to the increased complexity of the relationship that has to be approximated by the metamodel. This is supported by the experience of Martinez et al. (2007), who developed an ANN metamodel for the Valencia WDS. Consequently, the aim of this paper is to extend the ANN-GA optimization methodology developed by Broad et al. (2005) to cater for more complex WDSs. The proposed approach is tested on the Wallan WDS in Victoria, Australia, which is more complex than systems to which the ANN-GA approach has been applied to previously (e.g. Salomons et al., 2007, Martinez et al., 2007), both in terms of network size and the inclusion of water quality considerations.

4.2 Proposed Methodology

4.2.1 Introduction

While it is relatively straightforward to obtain a satisfactory ANN approximation for simple WDSs, this is not necessarily the case for more complex systems. Consequently, it is beneficial to reduce the complexity of the relationship that has to be approximated by the ANN. In order to achieve this, a number of factors should be considered, including the complexity of the hydraulic simulation model, the

complexity of the decision space, and the locations at which outputs are required from the hydraulic simulation model. Each of these issues is discussed in more detail below.

4.2.2 Complexity of Hydraulic Simulation Model

The more complex the simulation model, the more complex the relationship that needs to be approximated by the ANN model. In addition, increased model complexity increases the time taken to generate the requisite ANN calibration (training) data. Consequently, careful consideration needs to be given to the degree of complexity that is required for the hydraulic simulation model. Factors that need to be considered include (i) the complexity of the pipe network, which can be reduced using techniques such as such as skeletonization (e.g. Haestad Methods, 2002), decomposition (e.g. Deuerlein, 2008) or Gaussian elimination (e.g. Ulanicki et al., 1996), (ii) the duration of the simulation, which needs to be greater than the water age, (iii) simulation resolution, which needs to be sufficiently fine to avoid a loss in accuracy and (iv) control duration, which needs to be large enough to minimize any numerical irregularities.

4.2.3 Complexity of Decision Space

As GAs generate solutions randomly, there is a danger that irrational solutions are obtained as part of the optimization runs (e.g. solutions can be generated in which the upper tank trigger levels are lower that the corresponding lower tank trigger levels). This is undesirable, as it not only increases the size of the search space, but also makes it more difficult to develop an accurate metamodel due to the increased complexity of the decision space. Consequently, as much *a priori* information as possible about the system being modeled should be utilized.

4.2.4 Locations at which Simulation Model Outputs are Required

As part of hydraulic simulation models, outputs (e.g. pressures, chlorine residuals) are obtained at all time steps and at every node. However, developing an ANN metamodel that would provide the same level of information would necessitate the inclusion of a node in the ANN output layer for each of the nodes in the EPANET model. Clearly, this level of complexity in the ANN model is undesirable, as it makes it more difficult, and increases the time required, to calibrate (train) the model. Consequently, it is desirable to identify a set of critical nodes at which ANN outputs are required. Initially, the set of candidate critical nodes, $\{Out_{Ca}\}$, is equal to the set of all nodes containing a specific constraint. In order to reduce this candidate set, the following five-stage statistical process is proposed, whereby candidate nodes are ultimately categorized into the critical set, $\{Out_{Cr}\}$, or the redundant set, $\{Out_{R}\}$. It should be noted that there is likely to be a different number of critical nodes for hydraulics ($N_{C\text{-}HYD}$) and water quality ($N_{C\text{-}WQ}$) outputs. The categorization of nodes uses a set of randomly generated solutions, known as the metamodel development data. After the following checks are carried out these same data are used for calibrating the ANN metamodel. Broad et al. (2005) recommend using 10,000 randomly generated solutions in order to achieve adequate coverage of the solution space.

Data Range Check

The metamodel development data should be checked to ensure that there is variation in the data for each of the candidate output nodes. If there is no variation, the development of a metamodel is unnecessary. The candidate output is then deemed redundant, as shown in Eq. 4.1.

$$X \in \{Out_R\}\ if\ \max_{i \in ND}(V_{X_i}) - \min_{i \in ND}(V_{X_i}) < \varepsilon_{RANGE} \tag{4.1}$$

Where V_{X_i} is the *i*-th value of candidate output *X* (e.g. simulated pressure head (m), or chlorine residual (mg/L)); *ND* is the sample size of the metamodel development data used; and ε_{RANGE} is a user-defined threshold.

Demand Check

Candidate output nodes should be checked to determine whether they have a demand. If there is no demand at a given node, it is inconsequential whether a constraint is satisfied or not and that node can be culled from the set of candidate outputs and deemed redundant, as shown in Eq. 4.2.

$$X \in \{Out_R\}\ if\ DM(X) = 0 \tag{4.2}$$

Where $DM(X)$ is the average demand of output node X over the control duration.

Dominance Check

In the set of metamodel development data, if the magnitude of failure of candidate output *Y* is always greater than that of candidate output *X*, then output *Y* is said to dominate output *X* and output *X* may be deemed redundant. This is expressed mathematically in Eq. 4.3.

$$X \in \{Out_R\}\ if\ F_{Y_i} \geq F_{X_i}, \forall i \in ND \tag{4.3}$$

Where F_Y and F_X are the magnitudes of failure for outputs *X* and *Y*, respectively, where the magnitude of failure is defined in Eq. 4.4.

$$F_{X_i} = \max(0, V_{X-min} - V_{X_i}) \tag{4.4}$$

Where $V_{X\text{-}min}$ is the minimum allowable value corresponding to *X*.

This type of behavior is likely to occur more frequently for constraints on chlorine residuals and pressure heads with little variation in elevation. Upstream nodes in the WDS will generally have higher pressure heads and chlorine residuals and will therefore be more likely be dominated by downstream nodes.

Correlation Check

If the correlation coefficient of the metamodel development data between two candidate outputs is sufficiently high, one of the outputs can be deemed redundant. A high correlation is likely to occur where two nodes are spatially close, and the relative influence of each decision variable is similar for each of the candidate outputs. The correlation between two candidate outputs must be greater than a minimum correlation threshold, θ, for one output to be deemed redundant. The selection of an appropriate threshold value is important; if the threshold is set too high, too few candidate outputs will be made redundant and the number of critical outputs will be too high; conversely, if the threshold is set too low, outputs that should have associated metamodels trained for them will be made redundant. This will be problematic at the deployment stage, when the metamodel is acting as a surrogate for the simulation model. A solution presented to the metamodel may be deemed a good solution with no penalty costs, but that could be because one of the constraints has not been checked, because there was no output for it in the metamodel.

Once it has been determined that two candidate outputs are correlated, the next step is to select which output is placed in the redundant set. Recall that this approach assumes that for a given solution, only the worst node is penalized. Therefore, the candidate output with the lowest average magnitude of failure should be placed in the redundant set. This step is expressed mathematically in Eq. 4.5.

$$X \in \{Out_R\}\ if\ corr(F_X, F_Y) > \theta\ and\ \underset{i \in ND}{\text{avg}}\left(F_{X_i}\right) < \underset{i \in ND}{\text{avg}}\left(F_{Y_i}\right) \tag{4.5}$$

Frequency of Criticality Check

The preceding four steps will have resulted in a reduction in the set of candidate outputs to a set where each of the remaining outputs has solutions which have failed at least once in the *ND* metamodel development data solutions. However, it may be the case that some nodes are critical so infrequently that there is no need to keep them in the critical node set. Eq. 4.6 shows that candidate output X should be moved to the redundant set if its frequency of failure is below some threshold, ε_{CRIT}.

$$X \in \{Out_R\}\ if\ \frac{\sum CC_C}{ND} < \varepsilon_{CRIT} \tag{4.6}$$

Where CC_X is the criticality indicator function, given by Eq. 4.7.

$$CC_X = \begin{cases} 1 & if \quad F_{X_i} = \min\limits_{\{Out_{Ca}\}}[F_i] \\ 0 & otherwise \end{cases} \tag{4.7}$$

This step should be used with caution, as even nodes that are only critical infrequently might still be important when performing an optimization. As such, the value of ε_{CRIT} should be set fairly low. Regardless, if, during an optimization run, the GA converges to a part of the search-space that violates a node without a corresponding metamodel output, that node should be added back into the set of critical nodes.

4.2.5 Summary of Proposed Methodology

The proposed methodology for the optimization of WDSs using ANN metamodels that caters for WDSs with realistic levels of complexity is summarized below. The procedure is an extension of that introduced by Broad et al. (2005) and

incorporates the approaches to dealing with complex WDSs discussed in previous sub-sections.

1. Formulate the optimization problem (including constructing the objective function, and determining constraints and decision variables).
2. Develop ANN metamodel:
    a. Develop simulation model. Check for appropriateness of (i) model complexity, (ii) simulation duration and resolution, (iii) control duration and (iv) complexity of the decision space.
    b. Randomly generate metamodel development data using the simulation model, ensuring that only hydraulically plausible solutions are generated (e.g. no negative pressures);
    c. Determine critical nodes at which simulation model output is required;
    d. Calibrate (train) the ANN models(s) using the back-propagation algorithm after dividing the metamodel development data into training, testing and validation sub-sets;
3. Solve the optimization problem:
    a. Optimize the problem using the trained ANN(s) in place of the simulation model. During the optimization;
        i. Evaluate every new best solution with the simulation model (as a metamodel is only an approximation, solutions obtained by the GA must be checked against the original simulation model to ensure feasibility);
        ii. Keep track of the set of best "x" solutions, according to the ANN(s) (the best solution according to the simulation model

may correspond to the second, third, tenth (etc) best solutions according to the metamodel);

b. Conduct a local search using the simulation model, commencing from the best solution from Step 3.a. (again, because the simulation model is only an approximation, small improvements in the quality of the solution can be achieved by using a local search after running a GA).

It should be noted that although a number of uncertainties are introduced into the ANN development process by following the above approach (e.g. simplification of hydraulic model, ANN data reduction), resulting in inaccuracies in the ANN metamodel, these have a minimal effect on the outcome of the optimization. This is due to steps 3a. and 3b. above, which cater for the errors in the metamodel (e.g. by relaxing the constraints) and ensure that optimal feasible solutions obtained using the ANN metamodel are checked against solutions obtained using the actual simulation model.

4.3 Case Study: Wallan

4.3.1 Introduction

The purpose of this case study is to determine the optimal way in which the water supply system for the town of Wallan, which is located near Melbourne, Australia, should be operated in the short term, assuming average summer demands. In order to achieve this, the ANN-GA based optimization approach introduced in the previous section was applied to the system.

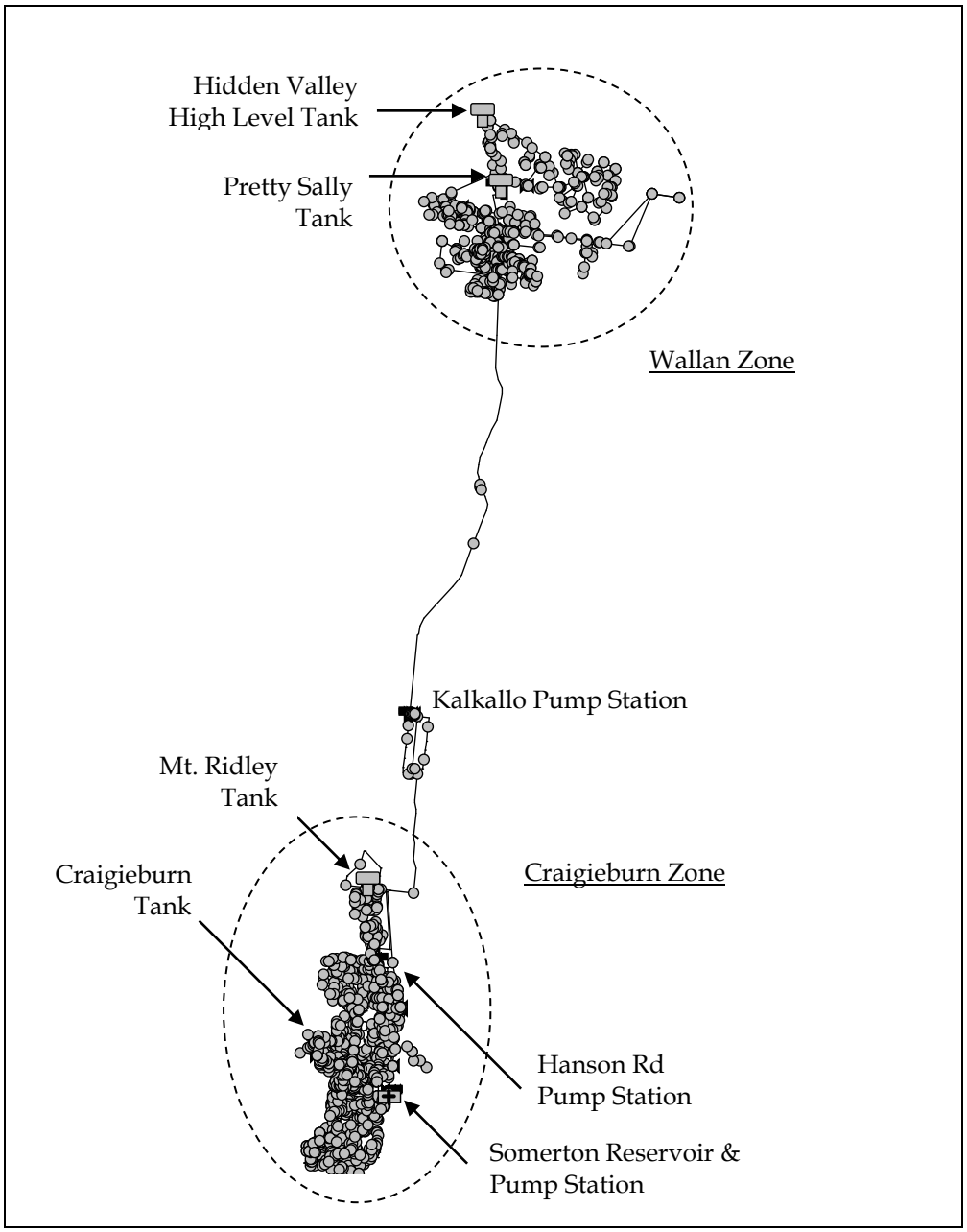


Figure 4-1. Layout of Wallan case study area.

The layout of the WDS under consideration is shown in Fig. 4.1. The water source is in the south at the Somerton reservoir. Water is pumped to the Craigieburn tank from Somerton and also pressurizes the large Craigieburn zone when in operation. Water is then pumped up to Mt Ridley tank from the Craigieburn zone via Hanson Rd pump station. Kalkallo pump station draws water from Mt Ridley tank to supply the Wallan zone via a newly constructed 450mm main into the Pretty Sally

tank. An outlet main from Pretty Sally tank feeds a pump station that boosts to Hidden Valley high level tank.

To test the various steps involved in the proposed approach, a series of scenarios were conducted using hydraulic simulation models with different levels of complexity (degree of skeletonization) and resolution (control duration) and using different levels of *a priori* system knowledge about the decision space to generate the ANN metamodel development data. This enabled a critical assessment of the importance of the different components in the overall methodology to be made. The four metamodeling scenarios considered are detailed in Table 4.1.

Table 4.1. Different ANN metamodel scenarios considered.

ANN Metamodel	**Degree of Skeletonization**		**Control Duration**		**Decision Variable Generation**	
Scenario	**Original**	**Skeletonized**	**24h**	**168h**	**Without *a priori* knowledge**	**With *a priori* knowledge**
1	X		X		X	
2	X		X			X
3	X			X		X
4		X		X		X

4.3.2 Problem Formulation

Problem formulation involves definition of the options that are available for operating the WDS (decision variables), the constraints that have to be satisfied and the objectives that are being optimized.

Decision Variables

The case study included two types of decision variable; tank trigger levels for switching pumps on and off, and chlorine dosing rates. There were 10 pumps and one

valve across 4 different pump stations. Hydraulic control was implemented through the use of a set of rules. Each control rule consisted of a pump or valve, controlling tank, tank level at which the pump or valve is switched on or off, and a time of day (peak or off-peak electricity tariff). The range of possible values for these decision variables is given in Table 4.2. For 11 pumps and valves, 2 electricity tariffs, and 2 trigger levels, there were ($11 \times 2 \times 2$ =) 44 decision variables.

Table 4.2. Allowable range for tank trigger levels.

Station	Infrastructure Type	No. of Pumps /Valves	Controlling Tank	Tank Trigger Levels			
				Min	Max	Res	Options
Hanson Rd	Valve/Pump	3	Mt. Ridley	0	8	0.1	81
Kalkallo	Pump	2	Pretty Sally	0.1	3.9	0.1	39
Pretty Sally Outlet	Pump	2	Hidden Valley	0.5	9	0.1	86
Somerton	Pump	4	Craigieburn	0	12.6	0.1	127

In addition to the hydraulic decision variables, there were 5 chlorine dosing locations. Four of these chlorinators were already in operation, while a fifth, at Kalkallo, was an option being considered as part of this optimization study. Possible chlorine dosing rates ranged from 0-3 mg/L, in increments of 0.1 mg/L.

Therefore there were a total of 49 decision variables and the size of the search-space was ($31^5 \times 2 \times 81^3 \times 39^2 \times 86^2 \times 127^4$) = 8.9×10^{28}.

Constraints

Hydraulic constraints were placed on critical nodes in the system that exceeded acceptable minimum pressure heads. These acceptable minimum pressure heads were set at the current (pre-optimization) values; that is, the constraint was to ensure the optimization did not result in lower pressures anywhere in the system

than are experienced currently. These values were calculated with the aid of the EPANET model. The minimum allowable residual chlorine concentration was set to 0.1 mg/L throughout the system. Penalty multipliers for pressure heads and chlorine concentrations were set at $\$10^7$/m and $\$10^7$/mg/L, respectively, as suggested by Broad et al. (2005) to ensure that the GA does not converge to infeasible values.

Objective Function

The objective function consisted of two components for material cost (energy and chlorine) and two penalty costs (pressure and chlorine residual). Energy costs were associated with operating existing pumps. Chlorine costs consisted of two parts; capital and on-going. The capital cost was for installing a new chlorinator at Kalkallo (if the optimization deemed it necessary). The on-going costs included the cost of maintaining both existing and new chlorinators, as well as the cost of chlorine (as hypochlorite).

The daily chlorine dosing cost for the *i*-th dosing location is given by:

$$CC_{DAY}^{i} = CC_{UNIT} \sum_{t=0}^{T_{CONTROL}} Q_t^i C_{0,t}^i + CC_{MNTNC}^i \tag{4.8}$$

Where CC_{UNIT} is the unit cost of chlorine in \$/kg; Q_t^i is the flow through the *i*-th chlorinator at time *t*; $C_{0,t}^i$is the chlorine dosing rate for the *i*-th chlorinator at time *t*; $T_{CONTROL}$ is the control duration for which the system is optimized; and CC_{MNTNC}^i is the daily maintenance cost of the *i*-th chlorinator.

The capital cost of the new chlorinator is given by:

$$CC_{CAPITAL} = \begin{cases} 250{,}000 & if \quad C_0^{KALKALLO} > 0 \\ 0 & otherwise \end{cases} \tag{4.9}$$

Where $C_0^{KALKALLO}$ is the set point for the potential new chlorinator at Kalkallo.

The total chlorine cost is therefore given by:

$$CC_{TOTAL} = CC_{CAPITAL} + \sum_{i=1}^{N_C} NPV\left(365 \times CC_{DAY}^{i}, r, T_{DESIGN}\right) \tag{4.10}$$

Where *NPV(·)* is the net present value function; r is the discount rate; T_{DESIGN} is the design horizon, which was selected to be 25 years; N_C is the number of chlorinators.

The daily energy cost is given by:

$$EC_{DAY}^{i} = EC_{P,UNIT} \sum_{t=0}^{T_{CONTROL}} \left(E_{P,t}^{i}\right) + EC_{OP,UNIT} \sum_{t=0}^{T_{CONTROL}} \left(E_{OP,t}^{i}\right) \tag{4.11}$$

Where $EC_{P,UNIT}$ and $EC_{OP,UNIT}$ are the unit energy costs for the peak and off-peak times of the day, respectively; $E_{P,t}^{i}$and $E_{OP,t}^{i}$ are the amount of energy consumed for pump *i* in time *t* for the peak and off-peak times of the day.

All pumps are already in place, therefore there are no capital costs for new pumps. Also, maintenance costs for pump stations have not been included, as they would not affect the optimal solution. Maintenance costs will simply add a fixed cost to whatever solution is selected. Therefore, the total energy cost is given by:

$$EC_{TOTAL} = \sum_{i=1}^{N_P} NPV\left(365 \times EC_{DAY}^{i}, r, T_{DESIGN}\right) \tag{4.12}$$

Where N_P is the total number of pumps in the system.

Hydraulic and water quality penalty costs are given as PC_{HYD} and PC_{WQ}, respectively, and correspond to the aforementioned constraints. Combining actual and penalty costs gives the following overall objective function:

$$TC = CC_{TOTAL} + EC_{TOTAL} + PC_{HYD} + PC_{WQ} \tag{4.13}$$

Values for the fixed variables in Equations 4.8-4.13, as supplied by the water authority operating the system (Yarra Valley Water (YVW)), are given in Table 4.3.

Table 4.3. Objective function parameter values.

Parameter	Value	Units
CC_{UNIT}	1.79	$/kg
CC^{i}_{MNTNC}	100	$/day
$EC_{P,UNIT}$	4.9	c/kWh
$EC_{OP,UNIT}$	2.6	c/kWh
r	10	% p.a.
T_{DESIGN}	25	Years
$T_{DURATION}$	24	Hours

4.3.3 Development of ANN Metamodels

As was the case with the New York Tunnels ANN-GA case study conducted by Broad et al (2005), there are two categories of ANNs; hydraulic and water quality. The hydraulic ANNs have the 44 tank trigger levels as inputs. The outputs include minimum pressure head over the control duration at the $N_{C\text{-}HYD}$ critical nodes, as well as energy consumed (peak and off-peak). The water quality ANNs have the trigger levels as inputs, as well as the 5 chlorine dosing rates. Outputs include the minimum chlorine residual at the $N_{C\text{-}WQ}$ critical nodes, and the total mass of chlorine dosed

throughout the system. This final output is required to calculate the chlorine cost. Chlorine dosing is in units of mass per volume and will depend on the flow at different times of the day and for different solutions, the total chlorine dosed will not be constant. Therefore, the metamodel will be required to approximate this value, as one could not obtain the flows without conducting an EPANET simulation. A schematic of the ANN metamodels is given in Figure 4-2.

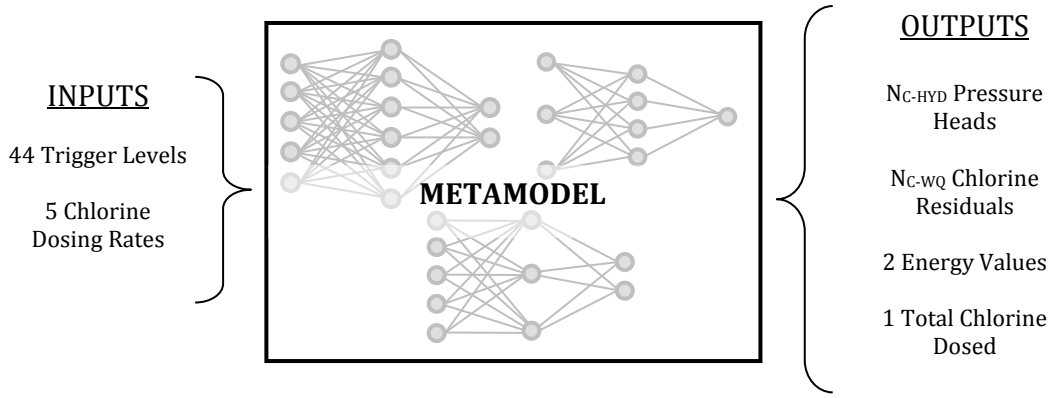


Figure 4-2. Metamodel structure for Stage 1 of HGC Case Study.

There are a number of different metamodeling approaches that may be used; a single ANN with several outputs, or many ANNs with one output each. It was decided that the latter option was more appropriate for this case study, as calibrating an ANN model for a single output generally improves predictive performance. While this is the more computationally demanding option, the additional time required will be small in comparison with the time required for generating the metamodel development data.

Development of Simulation Model

The EPANET model of the system provided by YVW was an "all-pipes" model, with 6-minute time-steps for both hydraulics and water quality. The demand patterns

were also in 6-minute time steps, with a duration of 5 days. The network consisted of more than 2000 pipes and 1700 nodes.

Degree of Simplification

In order to assess the impact of the degree of skeletonization on the ability of the ANN metamodel to approximate the simulation model, models with two different levels of complexity were considered (Table 4.1). As mentioned above, the original model was an "all-pipes" model, where many of the demand nodes represented areas as small as several houses. The problem with using a model with this level of detail is that accuracy cannot be as great at such small a scale. For example, the diurnal demand profile that is used is based on a spatial average of all demand nodes across the system. However, at the street level, actual demand would be much more sporadic, rather than change gradually. This could potentially lead to numerical instabilities and, therefore, poor metamodel performance.

Therefore, the model was skeletonized to remove some of these small scale effects. This was done using pipe diameters as the criteria to determine which pipes should be removed. Pipes with diameters of less than 100mm were removed, with the exception where this would result in the removal of entire sections of the system, in which case, some of the smaller pipes were retained. Hence, there was some degree of engineering judgment required. Nodes that were subsequently removed due to the skeletonization process had their respective demands aggregated to the nearest node. In addition, as many of the smaller pipes were removed, the minimum allowable chlorine residual was raised from 0.1 mg/L to 0.15 mg/L based on advice from YVW.

Table 4.4 presents a comparison of the complexity of the original and skeletonized models. It can be seen that approximately 34% of the pipes were

removed, along with 26% of the junctions and 64% of the loops. In order to validate the skeletonized model, an extended period simulation (EPS) was conducted with the skeletonized model and the results obtained were compared with those obtained from an EPS conducted with the original model. The metric used to ensure the accuracy of the skeletonized model was the average flow exiting the Somerton Reservoir. There was little difference in average flows (approximately 0.5%; 195.5 L/s compared with 194.3 L/s), thus validating the skeletonization process. Reducing the number of these loops had the associated benefit of reducing run-time by 68 seconds, or 59%, on average for a single evaluation.

Table 4.4. Comparative statistics between the original model and the skeletonized model.

Component	Model	
	Original	Skeletonized
Junctions	1730	1271
Reservoirs	1	1
Tanks	4	4
Pipes	2097	1376
Valves	35	34
Pumps	10	10
Loops	407	144
Run-time [s]	114	46

Simulation Duration and Resolution

The required simulation duration must be greater than the water age in the system to ensure the effect of chlorine dosing is sensed at the extremities of the network. The simulation duration must also be longer than the control duration so that costs and constraints can be calculated correctly. In addition, the simulation

duration should ideally be as short as possible to avoid excessive run-times. An analysis of the Wallan model indicated a maximum water age of 700 hours. Consequently, the simulation duration was set to this value. In addition, hourly demand patterns with a duration of 24h were used and the hydraulic time step was set at 1 hour. In contrast, a water quality time step of 6 minutes was used.

Control Duration

Analysis of the model showed that, despite 24 hour demand patterns being used for all demand nodes, when simulated in EPANET, the resultant repeating pattern for pressure head was sometimes longer than 24 hours. This is problematic, because it can result in noise being introduced into the metamodel development data, as the control duration is used to calculate values for constrained variables (pressure head and chlorine residual) and elements of the objective function, such as energy. Consequently, the impact of using two different control durations, 24h and 168h (7 days), was investigated (see Table 4.1). For the extended control duration, energy costs were calculated using the average values over 7 days. In addition, the pressure head and chlorine residual constraints were changed to use the minimum value of the final 7 days of the simulation, rather than the final 24 hours.

Generation of ANN Model Development Data

The EPANET models of different complexity and with different control durations were then used to generate the training, testing and validation data for the various ANN modeling scenarios considered (see Table 4.1). For each scenario, 10,000 data points were generated randomly for ANN model development purposes, as suggested by Broad et al (2005).

In order to assess the impact of the complexity of the decision space on ANN model performance, two types of ANN model development data were generated. In the first type, values of the decision variables (ANN inputs) were generated without consideration of any *a priori* system knowledge. In the second type, *a priori* knowledge about the operation of the system was used to constrain the generation of the ANN input data so that only physically plausible decision variable combinations were obtained (see Table 4.1).

In order to generate the second data type, the way the tank trigger levels were generated was changed. This involved three steps. The first step involved placing, or raising, minimum allowable operating levels. This overcomes the problem that solutions could be randomly generated in which tanks were allowed to empty before pumps would switch on etc.

Secondly, the trigger level decision variables were re-formulated so that they were grouped by pump station and hydraulically more sensible. For example, the N trigger level decision variables at a pump station were originally generated completely randomly and independently. The re-formulated approach involved generating N possible trigger levels within the feasible limits and then sorting them to ensure all upper trigger levels were higher than lower trigger levels.

The final adjustment to the tank trigger level formulation was specific to the Hanson Road pump station. When switched on, the pumps draw water from the Craigieburn tank and Somerton reservoir, and deliver water into the Mt Ridley tank, while also pressurizing a small section of the Craigieburn Zone. When the pumps are switched off, the Mt Ridley tank draws down, gravity feeding through the valve.

Hence, since the valve and pumps work in a complementary fashion, it was decided that decision variables associated with the valve should be removed.

Due to hydraulic issues, it was discovered that for the Hanson Road pump station, the lower trigger levels needed to be the same for both pumps, but that they could have different upper trigger levels. The valve would close when the tank level reached the lower trigger level (and the pumps would turn on); and the valve would open when the water level reached the higher of the two upper trigger levels. This resulted in a reduction of six in the number of decision variables.

Determination of Critical Nodes

The five-step procedure for determining critical nodes introduced in this paper was applied to each of the four ANN model development datasets generated in accordance with the scenarios outlined in Table 4.1. This determined the number of ANN models that had to be developed for each scenario, as a separate ANN was developed for each metamodel output, as discussed previously. The number of critical pressure and water quality nodes obtained is summarised in Table 4.5. It can be seen that the proposed procedure resulted in a significant reduction in the number of critical nodes and that the number of critical nodes obtained for the various datasets were very similar, ranging from 10 to 13.

Table 4.5. Critical nodes for different ANN metamodeling scenarios.

ANN Model Scenario	Number of Initial Nodes	Number of Critical Nodes Indentified		
		Pressure	Water Quality	Total
1	1730	4	6	10
2	1730	6	7	11
3	1730	5	7	12
4	1271	7	6	13

As discussed previously, the number of critical nodes identified is dependent upon several threshold values. The chosen threshold values were all quite conservative, however, for some applications, tests may need to be conducted to determine appropriate threshold values. If, during an ANN-GA run, the GA begins to exploit a particular node in the model because there is no ANN output (and hence no constraint being applied) the threshold values may need to be adjusted to increase the number of critical outputs.

The data range check threshold value, ε_{RANGE}, was set at a very small (conservative) value of 0.001, as the aim of this check is to detect "dead ends" in the model where values do not change with respect to the decision variables. The correlation check threshold, θ, was set to a high (conservative) value of 0.99 based on the results of a limited sensitivity analysis. However, the results of the sensitivity analysis showed that the number of critical nodes retained was relatively insensitive to this value. The frequency of criticality threshold value, ε_{CRIT}, was set to a very small (conservative) value of 0.001, which corresponds to 10 solutions in 10,000.

Details of the impact of each of the five steps for identifying a set of critical nodes are given in Table 4.6 for one of the four scenarios (scenario 4). It can be seen that the most significant reduction in the number of critical nodes for both hydraulics (pressure head) and water quality (chlorine residual) resulted from the correlation check. This is most likely due to the model containing many nodes in close proximity to each other, which are therefore highly correlated.

Table 4.6. Impact of different stages of proposed critical node determination procedure for scenario 4.

Critical Input Determination Stage	Remaining Critical Nodes	
	Hydraulics	**Water Quality**
Original	1271	1271
Data Range Check	1201	1262
Demand Check	851	909
Dominance Check	851	909
Correlation Check	174	40
Frequency of Criticality	7	6

Calibration and Validation of ANN Models

The generated metamodel development data were divided into training (80%), testing (10%) and validation (10%) subsets. In comparison to other applications (e.g. Bowden et al. 2002) a higher proportion of data was placed in the training set because the data were (a) expensive to generate, and (b) not noisy. The most appropriate proportions for the three sets might be different for different case studies, depending on the number of inputs to the metamodel and the amount of development data that may be generated in a reasonable timeframe. Hence, this is a potential area of future research.

The training data were used to adjust the ANN model parameters (weights) using the backpropagation algorithm. A series of training runs was performed with different values of learning rate, momentum rate and number of hidden nodes. The learning rate was varied between 0.1 and 0.5; the momentum rate between 0.2 and 0.5; and the number of hidden nodes between 10 and 60. This gave a total of 32 parameter combinations that were tested for each ANN model. The test data were

used to decide when to stop training using cross-validation and which combination of learning rate, momentum rate and number of hidden nodes resulted in optimal model performance. Finally, the validation data were used to check the performance of the selected model on an independent data set.

4.3.4 Results and Discussion

Metamodels were developed for the four scenarios shown in Table 4.1. The ANN modelling scenario that resulted in the best ANN model performance was linked with a GA model to solve the optimization problem for the Wallan WDS. A total of 30 optimization runs were conducted with different random number seeds in order to minimize any impacts due to the random starting position in search space. Optimization was conducted using a GA with integer coding, one-point crossover and a tournament size of two. Values of the GA parameters were selected by trial-and-error to obtain the best performance without excessive computational time and are given in Table 4.7. Further details on how the GA parameters were determined are given in Broad et al. (2005). The local search algorithm that was used is known as Sequential Downward Mutation (SDM), which was developed specifically for the purpose of WDS optimization. SDM was selected from a set of potential algorithms upon testing each on a benchmark case study (see Broad et al. 2006). The computational efficiency of the algorithm could be improved further by adopting a more sophisticated hybrid approach (e.g. Espinoza and Minsker 2006). However, this is unlikely to be a significant issue in this case, as the metamodeling approach speeds up the GA dramatically, thereby enabling the GA to be run to convergence before the local search is applied.

Table 4.7. GA Parameters used for the Wallan Case Study.

Parameter	Value
GA Type	integer
Population Size	400
Probability of Crossover	0.8
No. Crossovers per Pair	1
Probability of Bit-Wise Mutation	0.02
No. Generations	2000

ANN Metamodel Performance

The performance of the ANN models for the various scenarios considered (Table 4.1) is given in Tables 4.8 and 4.9. As can be seen, root mean squared (RMS) error and the coefficient of determination (R^2) were used as performance measures. It should be noted that the performance measures for the pressure head and chlorine residual predictions given are averaged over the critical pressure and water quality nodes, respectively (Table 4.5). Also, the results presented are the average values across the 32 different parameter combinations. However, it should be noted that ANN performance was not sensitive to the parameters chosen. Of all the metamodel outputs, the output with the greatest variance was the chlorine residual at one of the nodes in the Craigieburn area; it had a coefficient of variation of 0.039 over the 32 parameter combinations.

Table 4.8. Average RMS errors of the validation set for the various ANN model development scenarios considered.

ANN Output		Units	1	2	3	4
Pressure Head (ave)		[m]	3.728	0.391	0.433	0.110
Chlorine Residual (ave)		[mg/L]	1.053	0.244	0.213	0.017
Energy	Peak	[kWh]	1278.1	516.1	425.4	81.0
	Off-Peak	[kWh]	602.1	401.1	305.7	78.3
Chlorine Dosed		[kg]	3.240	2.170	1.800	0.590

Table 4.9. Average R2 values of the validation set for the various ANN model development scenarios considered.

ANN Output		1	2	3	4
Pressure Head (ave)		0.602	0.780	0.744	0.978
Chlorine Residual (ave)		0.556	0.679	0.654	0.996
Energy	Peak	0.504	0.635	0.742	0.983
	Off-Peak	0.641	0.711	0.812	0.982
Chlorine Dosed		0.967	0.984	0.991	0.999

It can be seen that the strategies suggested as part of the methodology introduced in this paper have a significant impact on ANN metamodel performance. The worst performance was obtained when the more complex EPANET model was used in conjunction with the shorter control duration and the random generation of tank trigger levels (scenario 1, Table 4.1). The elimination of physically implausible combinations of tank trigger levels (scenario 2, Table 4.1) resulted in a significant improvement in the prediction of all ANN outputs. This is because of the resulting simplification of the input-output relationship that has to be estimated by the ANN model.

An increase in the control duration from 24h to 168h to allow a repeating pattern of pressure head to be established (scenario 3, Table 4.1) had a moderate effect on some ANN model outputs, such as energy consumption and chlorine dosage, but had very little impact on the predictions of pressure head and chlorine residuals at the critical nodes. Finally, the skeletonization of the model (scenario 4, Table 4.1) had a significant impact on the ability of the ANN models to predict all of the output variables, further highlighting the importance of simplifying the relationship to be estimated by the ANN as much as possible. The performance of the ANN model developed as part of scenario 4 was excellent, with very low RMS errors and very high R^2 values. RMS errors of 0.1m for pressure head are quite low relative to the range of values in the metamodel development data, which was greater than 4.6m for all critical nodes. Similarly, RMS errors of 0.02mg/L for chlorine residual are very low in when considering typical measurement tolerances are 0.05mg/L or higher (Phelps 2008). This model was therefore combined with the GA in order to solve the Wallan WDS optimization problem. Based on the trial and error approach described earlier, the parameter values for the selected ANN were as follows: learning rate: 0.4; momentum rate: 0.5; and number of hidden nodes: 60.

Optimization

Thirty optimization runs were conducted with different random number seeds. The minimum, average and maximum values obtained were \$1.34m, \$1.60m and \$1.71m, respectively. The variation in objective function values found is due to the local search that is employed at the completion of the ANN-GA run. By its nature, the local search can be prone to becoming trapped in local minima, and the quality of the solution found is dependent upon the starting position. The best solution of \$1.34

million represents a saving of 14% compared with the current operating regime with an estimated NPV of $1.56 million.

Details of the best optimal solution are presented in Table 4.10. The main cost component was the energy cost ($1.2 million), rather than the chlorine cost ($130,000). Recall that the chlorine cost comprised the capital cost of a new chlorinator to be built at Kalkallo (if selected), plus the NPV of the mass of chlorine dosed. The optimal solution did contain a pressure penalty but the value was trivially small, at $50,000, which was due to a violation of 0.05 mm. The optimal solution did not contain any a penalty for water quality.

Table 4.10. Summary of optimal solution obtained.

Solution Component	Current Operations	Optimized Solution	Units
Total Material Cost	1.56	1.34	$million
Chlorine Cost	0.04	0.13	$million
Energy Cost	1.52	1.20	$million
Pressure Deficit	0	0.00005	m
Quality Deficit	0.1	0	mg/L

The optimal solution had significantly lower energy costs, but had higher chlorine costs compared with the current operating regime. However, while the current operating regime had low chlorine costs, the amount dosed throughout the system was inadequate, as indicated by the quality deficit of 0.1 mg/L, which shows that some nodes did not have any chlorine residual.

Table 4.11. Single day energy costs (peak and off-peak tariffs) for the optimal solution with a comparison to current operations.

Pump	Current Operations		Optimized Solution	
	Peak	Off-Peak	Peak	Off-Peak
Hanson Rd #1	$31.07	$22.80	$-	$7.33
Hanson Rd #2	$67.18	$15.99	$71.71	$33.38
Kalkallo #1	$33.35	$11.61	$-	$-
Kalkallo #2	$45.29	$15.85	$37.30	$20.00
Pretty Sally #1	$40.84	$20.05	$0.10	$12.59
Pretty Sally #2	$21.36	$11.47	$26.23	$13.17
Somerton #1	$17.22	$13.22	$20.40	$3.48
Somerton #2	$16.80	$13.18	$22.96	$3.90
Somerton #3	$16.66	$12.75	$37.73	$4.78
Somerton #4	$17.22	$13.44	$37.75	$10.65
Total	$306.98	$150.37	$254.17	$109.27

A breakdown of the pumping costs is given in Table 4.11, which shows the daily pumping cost for each pump at each pump station. The total daily pumping cost was $363, including $254 during peak tariff times and $109 for off-peak. It can be seen that only one of the pumps at Kalkallo was utilised. However, while one pump is sufficient for this case, the second pump will be utilized in the future as the area grows. Table 4.11 also provides a comparison with the costs for the current operating regime. The total daily energy cost is $457, including $307 during peak tariff times and $150 for off-peak. The results show that the optimized solution included more peak pumping at the Somerton pump station, but that this was offset by less pumping during peak periods downstream in the system (Hanson Rd, Kalkallo and Pretty Sally).

Chlorine dosing details are presented in Table 4.12. Overall, the GA selected to dose more chlorine compared with the current operating regime; which is as expected, given that there were deficits in the required chlorine residuals. It is important to highlight that the optimal solution included no dosing at Kalkallo. Recall that part of the objective function included the capital cost at Kalkallo ($250,000) but all other chlorinators were already installed, hence their capital cost did not need to be considered. Therefore, it is evident that the more optimal choice is to administer higher chlorine doses at the other locations and not to construct a new chlorinator at Kalkallo. The main chlorinator is located at Somerton, which has a set point of 1.1 mg/L. The chlorinator at the Pretty Sally outlet feeds much of the Wallan area and therefore has a relatively high set point at 1.2 mg/L. Conversely, the chlorinators at the Pretty Sally Pump Station and the outlet of the Hidden Valley tank serve smaller areas and hence their optimum set points are lower; 0.2 and 0.5 mg/L, respectively.

Table 4.12. Chlorine dose rates [mg/L] for optimal solution with a comparison to current operations.

Chlorinator	Current Operations	Optimized Solution
Somerton	0.4	1.1
Kalkallo	N/A	0
Pretty Sally Outlet	0.4	1.2
Pretty Sally Pump Station	N/A	0.2
Hidden Valley Outlet	0.4	0.5

Computational Issues

A summary of the computational requirements is presented in Table 4.13. It should be noted that the ANN development data were generated in parallel on a

2.4 GHz Intel Xeon CPU, whereas training and optimization runs were conducted on a serial computer. A full comparison between the results obtained using an EPANET-GA and the ANN-GA approach could not be conducted due to the large computational requirements of the former option. Consequently, the optimization time for the EPANET-ANN approach given in Table 4.14 is an estimate based on using a population size of 400 and 2000 generations; the same values used by Broad et al. (2005a).

Table 4.13. Computational requirements optimization (hours).

Component	EPANET	Metamodel
Data Generation	N/A	288
Training	N/A	32
Optimization	25333*	1.4

*Estimate, based on average simulation time for EPANET model.

As can be seen from Table 4.13, use of the ANN-GA approach makes determining the optimal operating strategy of a WDS of realistic complexity (1271 nodes, 1376 pipes) feasible, even when considering both pressures and water quality, with an overall run-time of 1.4 hours compared with an estimated run-time of 1056 days if EPANET was used as the simulation model. It should be noted that the vast majority of the computational overhead associated with the ANN-GA approach is for the development of the ANN metamodels, particularly in relation to the generation of the requisite model development data using EPANET. Once the ANN model has been developed, each optimization run only takes 1.4h to complete, making the approach suitable for optimizing operational settings on a regular basis.

4.4 Conclusions

This paper presents a methodology for applying metamodels with GAs for determining optimal operating strategies for hydraulically complex water distribution networks. The methodology involves careful review of the complexity of the hydraulic network and whether this can be simplified, the application of *a priori* knowledge to reduce the complexity of the decision space and a systematic reduction in the number of locations at which simulation model outputs are required. The application of these steps can greatly assist in the development of metamodels and increase the feasibility of applying optimization to complex WDS.

This research demonstrated the application of the metamodel-based optimization methodology developed for the optimal operation of a real water distribution system; Wallan, Victoria, Australia. The development of an ANN metamodel for a specific WDS is not a trivial matter. The methodology involves several steps in determining the most appropriate problem formulation and model parameters to use. Four different metamodelling scenarios have been presented here. The results showed that, for this case study, some skeletonization of the model was required to achieve adequately accurate metamodels.

The optimization results show a reduction in the daily pumping costs from $457 to $363 compared with the current operating regime; a saving of 21%. The net present value (NPV) over 25 years was used as the objective function, which included both pumping and chlorine costs. The current operating regime would have corresponded to an NPV of $1.56 million, while the optimized solution had an NPV of $1.34 million; a saving of 14%. In addition to these economic benefits, the optimized solution achieved adequate disinfection throughout the system, whereas the current

operating regime corresponded to deficits in chlorine residuals at several locations in the system.

The results presented here show that the proposed metamodelling methodology can be successfully applied to a realistically-sized case study. This is an extension of the results presented in Broad et al. (2005), which involved a simpler methodology being applied to a hydraulically simple benchmark case study.

Future research in the area of metamodel-based optimization of WDSs is required to extend the methodology from operational strategies to real-time operations. To achieve this, a method for accounting for initial conditions (e.g. tanks levels and chlorine concentrations) and varying demands needs to be developed.

4.5 Acknowledgments

The authors would like to thank Asoka Jayaratne and Chris Saliba from Yarra Valley Water for their technical input and data for the Wallan Case Study. The authors would also like to thank the Co-operative Research Centre for Water Quality and Treatment, based in Adelaide, Australia, and the Australian Department of Education, Science and Training for their financial support of this project.

Chapter 5

Publication 4: Data Uncertainty

"Command those who are rich in this present world not to be arrogant nor to put their hope in wealth, which is so uncertain, but to put their hope in God..."

1 Timothy 6:17 (NIV)

Statement of Authorship

Title of Paper	Systematic Approach to Determining Metamodel Scope for Risk-Based Optimization and its Application to Water Distribution Sytem Design.
Publication Status	Submitted for Publication
Publication Details	Broad, D. R., Dandy, G. C., and Maier, H. R. (2014). "Systematic Approach to Determining Metamodel Scope for Risk-Based Optimization and its Application to Water Distribution Sytem Design." *Environmental Modelling and Software*.

Author Contributions

Name of Co-Author	Darren Broad		
Contribution to the Paper	Conceptual and theoretical development, interpretation and analysis of results, manuscript preparation and corresponding author.		
Signature		Date	

Name of Co-Author	Graeme Dandy		
Contribution to the Paper	Project supervision and review of manuscript.		
Signature		Date	

Name of Co-Author	Holger Maier		
Contribution to the Paper	Project supervision and review of manuscript.		
Signature		Date	

Although the manuscript has been reformatted in accordance University guidelines, and sections have been renumbered for inclusion within this thesis, the paper is otherwise presented herein as published.

Abstract

Metamodels have proven be very useful when it comes to reducing the computational requirements of Evolutionary Algorithm-based optimization by acting as quick-solving surrogates for slow-solving fitness functions. The relationship between metamodel scope and objective function varies between applications, that is, in some cases the metamodel acts as a surrogate for the whole fitness function, whereas in other cases it replaces only a component of the fitness function. This paper presents a formalized qualitative process to evaluate a fitness function to determine the most suitable metamodel scope so as to increase the likelihood of calibrating a high-fidelity metamodel and hence obtain good optimization results in a reasonable amount of time. The process is applied to the risk-based optimization of water distribution systems; a very computationally-intensive problem for real-world systems. The process is validated with a simple case study (modified New York Tunnels) and the power of metamodelling is demonstrated on a real-world case study (Pacific City) with computational speed-ups of several orders of magnitude.

5.1 Background

Evolutionary algorithms (EAs) have become common practice in the optimization of water resources management (WRM) models due to their ability to find near globally optimal solutions amongst a very large number of possible options, and the fact that they can be coupled with simulation models that are often used to calculate fitness functions. While these simulation models can be quite powerful, one negative aspect is that they can be computationally intensive, as combining them with EAs might require 100,000 simulations or more.

In recent years, it has been recognized that, in order to fulfil their potential, EAs need to be applied to real case studies (Maier et al., 2014). While there has been some progress in this regard, it also raises a number of challenges, including how to best deal with uncertainty and the resulting increase in computational intensity (Maier et al., 2014).

5.1.1 Uncertainty

Murphy et al. (2009) (as cited by Maier et al. 2014) provide the following three broad categories of uncertainty: (i) data-related (e.g. being unable to precisely quantify the magnitude of a parameter that is known to have some effect in a system), (ii) model-related (e.g. not knowing which parameters from a set of likely candidates significantly affect a system's output), and (iii) lack of knowledge (e.g. complete ignorance, where little is known about which parameters affect a system). In well-defined WRM problems, such as those related to engineered systems, the types of data required and the model structure are often known with a high degree of certainty. The uncertainty arises in gathering the data for a specific instance of that

problem type. In that case, reasonable estimates of uncertainty can be made to account for the lack of complete/perfect knowledge.

A common way to incorporate uncertainty into a WRM optimization problem is to use risk metrics. Definitions of risk differ slightly between types of problems, but there are some broad definitions that are relevant generically, such as reliability (concerning likelihood of non-failure) and vulnerability (the consequence or impact of failure, should failure occur) (Hashimoto et al. 1982).

The most common way of calculating these metrics is via Monte Carlo Simulation (MCS). However, this increases optimization run-time by several orders of magnitude because each evaluation of an objective function requires *n* evaluations with the simulation model, where accuracy increases asymptotically as *n* tends to infinity. Long run-times are exacerbated in real case studies, to the point where they may become prohibitive.

5.1.2 Metamodelling

Maier et al. (2014) identify three broad methods to increase computational efficiency of EA-based optimization: metamodelling, parallel computing and heuristics. Each method has been shown to be effective, however, this paper focuses on the use of metamodels to improve computational efficiency of risk-based optimization of water resources problems.

A metamodel is a high fidelity approximation to a simulation model that can be used as a surrogate for the said model where it is used repetitively, such as during an EA-based optimization, or for sensitivity analyses (Blanning 1975). Because metamodels replace a simulation model with a mathematically simpler model, they provide significant computational speed-up when used in lieu of a simulation model.

Metamodels have proven to be useful tools for speeding up optimization in a range of water resources applications, including model calibration (Behzadian et al. 2009; Lingireddy and Ormsbee 1998; Mugunthan et al. 2005); distribution system design (Bi and Dandy 2013; Broad et al. 2005; Broad et al. 2006); distribution system operations (Broad et al. 2010; Martinez et al. 2007; Rao and Salomons 2007; Salomons et al. 2007); and groundwater remediation (Aly and Peralta 1999; Johnson and Rogers 2000; Yan and Minsker 2006; Yan and Minsker 2011). For a thorough review of water resources metamodelling applications, the reader is referred to Razavi et al. (2012a).

Metamodel usage in water resources may also be classified by the framework in which they are used. Razavi et al. (2012a) present four framework definitions: basic sequential framework (BSF); adaptive-recursive framework (ARF); metamodel-embedded evolution; and approximation uncertainty. Further research is required to determine which framework is best, and even whether a globally superior framework exists or if this is application-dependent. Such research is beyond the scope of this paper and the BSF has been adopted here due to its successful application in previous studies (Borgonovo et al. 2012; Broad et al. 2005; Broad et al. 2006; Broad et al. 2010; Khu et al. 2004; Lingireddy and Ormsbee 1998).

One benefit of the BSF is that it aims to develop a metamodel that is reasonably accurate across the whole search space. This is important when coupled with an EA because, even after an EA has converged to a specific area of the search-space, it still has the potential to consider/evolve candidate solutions that are far from the converged area. In contrast, if the metamodel is only accurate in a sub-section of the search-space (as is the case when using an ARF), a new candidate solution that is far

from the converged space may be considered feasible when it is not or infeasible when it actually is. This could result in convergence to infeasible solutions or at least slow-down convergence, as the algorithm must wait until the next phase of metamodel re-calibration.

Metamodel development using the BSF, as used in this paper and previous papers by the same authors, consists of the following steps (adapted from Razavi et al. (2012a)):

1. Generate metamodel calibration data through design of experiment (DoE) and evaluation with original simulation model;
2. Calibrate metamodel to fit across all the generated data. The data are split into separate sets to ensure over-fitting is avoided and generalisation is obtained;
3. The metamodel is substituted for the simulation model and an EA is used to optimize the problem; and
4. Recognising that even high-fidelity metamodels will not provide perfect representations of the simulation model and that the global optimum will differ when the metamodel is used to evaluate objective functions and/or constraints than when the simulation model is used (Jin 2005), some solutions are tracked as the EA runs and a local search is carried out post-optimization with the original simulation model.

Maier et al. (2014) identified that improved guidelines are required for metamodel development. One step of developing and using a metamodel is to determine its scope. For example, one question that needs to be answered is whether the metamodel should act as a surrogate for the whole fitness function, or a component of it. Both

approaches have been used in the literature. For example, in some studies, the whole fitness function (or at least one objective in multi-objective problems) is approximated (Bau and Mayer 2006; Jin et al. 2002; Khu and Werner 2003; Mugunthan et al. 2005; Ostfeld and Salomons 2005; Shoemaker et al. 2008), while in others, only a component of the fitness function is approximated, such as penalty functions used to account for constraint violation (Behzadian et al. 2009; Broad et al. 2005; Broad et al. 2006; Broad et al. 2010; Carnevale et al. 2012, Johnson and Rogers 2000; Kourakos and Mantoglou 2009; Yan and Minsker 2011; Zhang et al. 2009). If no thought is given to the scope of the metamodel or if this is done in an *ad hoc* manner, there is the risk that the metamodel cannot be calibrated to as high a fidelity as possible, and/or the benefits in terms of improved computational efficiency will not be fully realised.

It is likely that the best metamodel scope is problem-dependent and that researchers therefore generally consider which metamodel scope is most appropriate for the problem at hand. However, this is generally done in an *ad hoc* manner. Consequently, the focus of this paper is on (i) presenting a systematic approach to metamodel scope identification that can be used in any metamodel-based optimization application, thereby providing much needed guidance on the development of metamodels for increasing the computational efficiency of EA when applied to real-world water resources problems and ensuring that the developed metamodels are as accurate and computationally efficient as possible; (ii) illustrating the approach for the risk-based optimization of the design of water distribution systems, including a detailed analysis of the properties of the generic components of the fitness function and the identification of which of these components are best replaced by metamodels that is widely applicable to a variety of instances of this class

of problem; and (iii) the application of the proposed approach to two case studies, including a novel real-life system, for the risk-based optimal design of water distribution systems using EAs, considering both hydraulic and water quality performance, which has not been undertaken previously, thereby illustrating the benefits of the proposed approach. It should be noted that the application presents a novel contribution in itself, as previous applications of metamodelling to WDS optimization have not considered water quality and uncertainty, nor the second case study system.

5.2 Proposed Metamodelling Approach

The metamodelling framework proposed in this paper builds on the work presented in Broad et al. (2005) and Broad et al. (2010) by introducing an additional step that identifies which parts of the fitness function are amenable to being approximated using a metamodel (step 2), and modifying the algorithm that determines which solutions should be checked by the simulation model by recognising the computational budget is limited (step 8). The framework consists of the steps shown in Figure 5.1.

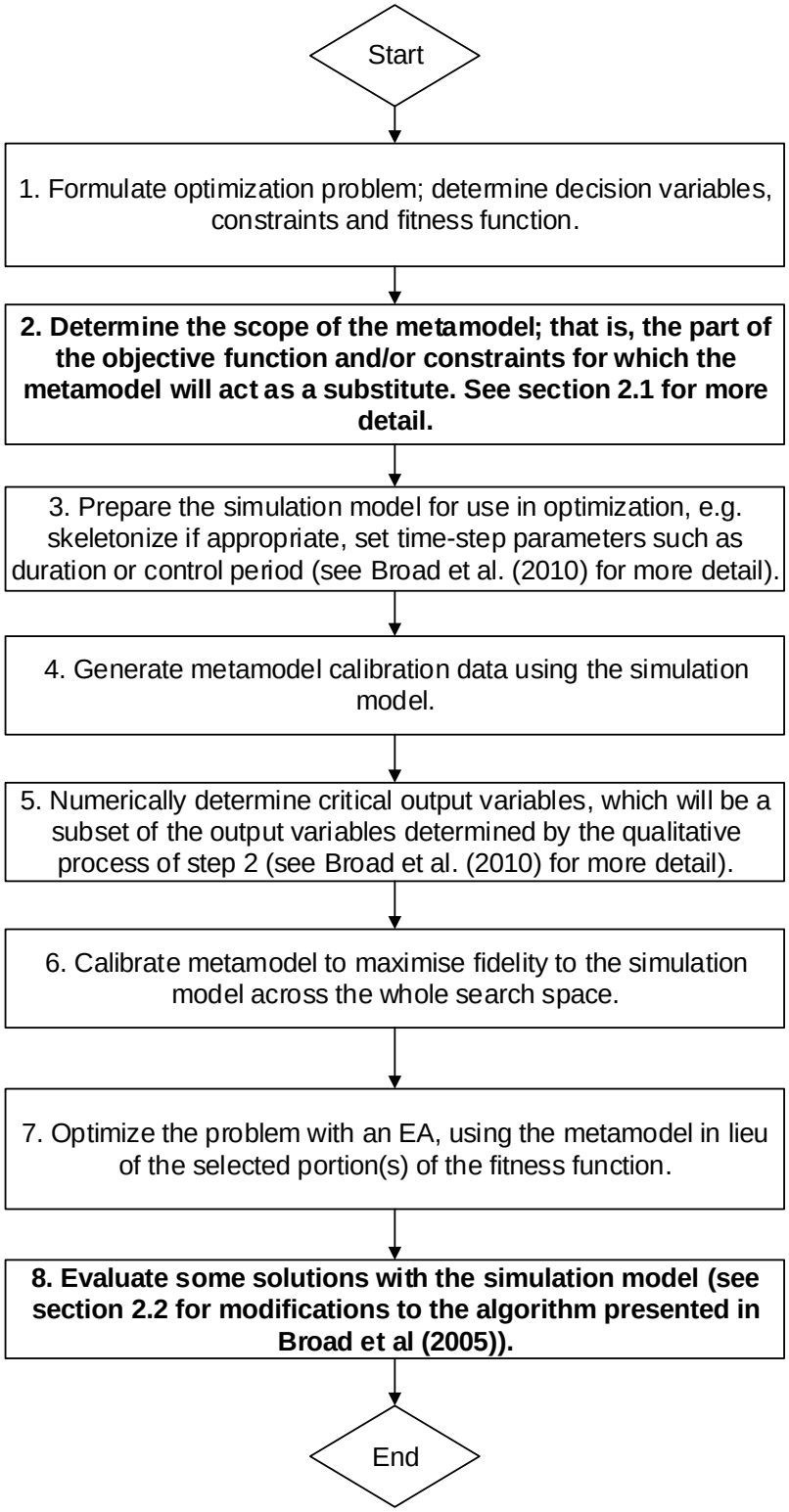


Figure 5-1. Basic Sequential Framework steps to develop and use a metamodel for EA-based optimization, with new/modified steps from this paper in bold.

Details of the novel steps of the above approach introduced in this paper (i.e. Steps 2 and 8) are given in the subsequent sections.

Note, this whole procedure is built on the assumption that the optimisation problem has already been assessed and that the computational demand of fitness evaluations is so high that the total estimated optimisation run-time is deemed unacceptable and hence metamodels may provide a benefit to reducing this overall optimisation run-time.

5.2.1 Metamodel Scope Definition (Step 2)

A fitness function, $z = f(\boldsymbol{x})$ can be broken down into a series of N intermediate calculation steps, f_i, i = 1,...,N, which may correspond to different objectives and penalties associated with constraints, as shown in Eq. 5.1.

$$\begin{aligned} \boldsymbol{y_1} &= f_1(\boldsymbol{x_1}) \\ \boldsymbol{y_2} &= f_2(\boldsymbol{x_2}) \\ &\dots \\ \boldsymbol{y_{N-1}} &= f_{N-1}(\boldsymbol{x_{N-1}}) \\ z &= f_N(\boldsymbol{x_N}) \end{aligned} \tag{5.1}$$

Where $\boldsymbol{x_i}$ is a vector of input variables to the calculation step, f_i, and whose dimensionality may be equal to, greater than, or less than the number of decision variables. The vector, $\boldsymbol{x_i}$, is by definition the concatenation of a subset of the decision variables, $\boldsymbol{dv}$, a subset of the set of external variables, $\boldsymbol{ev}$, and a subset of the intermediate calculated variables, $\boldsymbol{y}$, and is shown in Eq. 5.2.

$$\boldsymbol{x_i} \equiv \subseteq \{\boldsymbol{dv}\} \bigcup \subseteq \{\boldsymbol{ev}\} \bigcup \subseteq \{\boldsymbol{y}\}, \forall i = 1, \ldots, N \tag{5.2}$$

Where $\{\boldsymbol{y}\}$ is the set of all calculated variables, $\boldsymbol{y_i}$, i = 1,..., N.

This is demonstrated graphically in Figure 5.2, which shows how a generic fitness function is calculated from an optimisation algorithm's candidate solution (e.g. a Genetic Algorithm (GA) string).

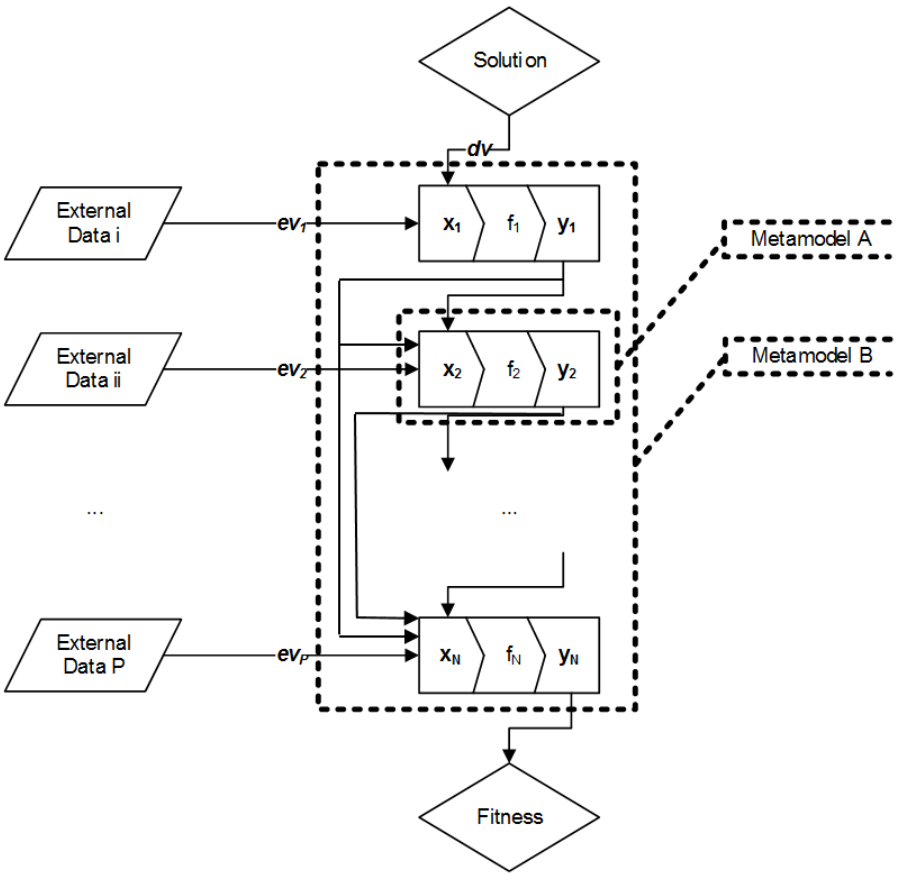


Figure 5-2. Generic fitness evaluation and two possible metamodel scopes.

A metamodel can act as a substitute for one or more "connected" calculation steps, where two calculation steps, *A* and *B*, are considered connected if the output of one comprises part of the input of the other, i.e., $\boldsymbol{y_A} \in \boldsymbol{x_B}$. So, generically, the scope of a metamodel can be defined by Eq. 5.3:

$$S \equiv \{f_i\} : \exists j \neq i : \boldsymbol{y_j} \in \boldsymbol{x_i} \tag{5.3}$$

The input variables to the metamodel, $\boldsymbol{x_{MM}}$, are then defined as the subset of all input variables that are inputs to the calculation steps that comprise the metamodel that are not also outputs from one of the other metamodel calculation steps (see Eq. 5.4).

$$\boldsymbol{x_{MM}} \equiv \subseteq \{\boldsymbol{x_i}\}, \forall i: f_i \in S, \boldsymbol{y_i} \in \boldsymbol{x_j}, \forall j: f_j \notin S \tag{5.4}$$

Similarly, the metamodel output variables, $\boldsymbol{y_{MM}}$, are the subset of all output variables that are outputs from the calculation steps that are not also inputs to the other metamodel calculation steps (see Eq. 5.5).

$$\boldsymbol{y_{MM}} \equiv \subseteq \{\boldsymbol{y_i}\}, \forall i: f_i \in S, \boldsymbol{y_i} \in \boldsymbol{x_j}, \forall j: f_j \notin S \tag{5.5}$$

The dashed lines in Figure 2 show two potential metamodels. The calculation points, f_i, within the metamodel are the calculations for which the metamodel will act as a surrogate. Therefore, for Metamodel A:

- the input variables, $\boldsymbol{x_{MM\text{-}A}}$, are comprised of the outputs of calculation step 1, $\boldsymbol{y_1}$, and external variables $\boldsymbol{ev_2}$;
- the metamodel replaces calculation step f_2; and
- the output variables are $\boldsymbol{y_2}$.

For Metamodel B:

- the input variables, $\boldsymbol{x_{MM\text{-}B}}$, are comprised of the decision variables, $\boldsymbol{dv}$, and all P external data sources, $\{\boldsymbol{ev}\}$;
- the metamodel replaces all N calculation steps $f_{1,..,N}$; and
- the output variable is the objective function, z.

For application to EA-based optimization, there are two principles that should drive a metamodel's development: (1) the computational speed-up it provides and (2) its fidelity to the original simulation model. So, when determining the calculation steps that should be replaced by a metamodel, they should (1) include the slower calculation steps, so as to maximize the benefits of metamodelling and (2) maximize fidelity (or not be too difficult to approximate). Two factors that are known to make developing high fidelity metamodels difficult are high dimensionality (Jin 2005, Caballero and Grossmann 2008 and Razavi et al. 2012a) and the presence of discontinuities in the function being approximated (Meckesheimer et al. 2001, Turner et al. 2003, Sasena et al. 2003 and Bauman 2013).

Based on these principles and experience of previous researchers, the following is a proposed qualitative process by which the best scope of a metamodel can be determined (see Figure 5.3 for a summary).

A metamodel may replace one or more of the calculation steps, $\boldsymbol{f}_i$, of a fitness function. Each calculation step should be assessed against the following three criteria in order to determine whether it should be included in the scope of the metamodel:

1. Computational Assessment: Consider the computation time for each calculation step. The metamodel should include all significant computationally expensive calculation steps.
2. Dimensionality Assessment: Consider the dimensionality of the metamodel, i.e. the number of inputs and outputs.
3. Smoothness Assessment: Consider whether there are any discontinuities present either in the function or its derivative, or whether the function is smooth.

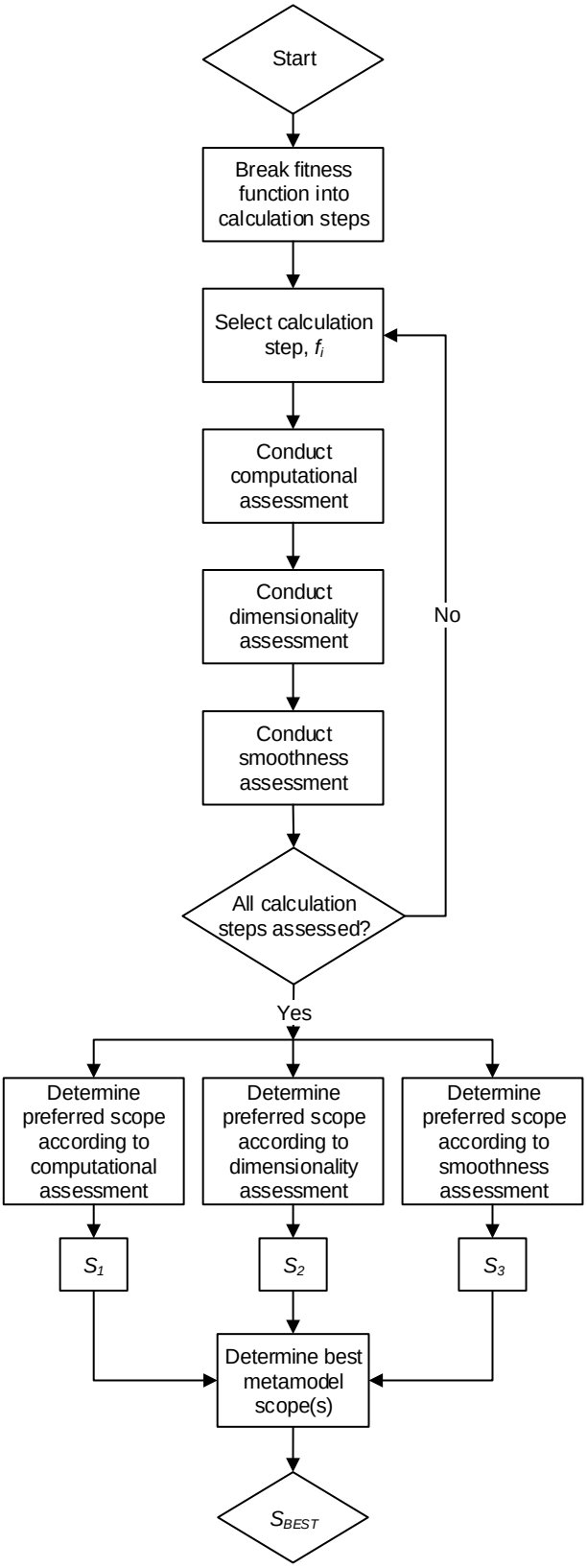


Figure 5-3. Proposed process for determining best metamodel scope.

The assessment against the three criteria must be carried out qualitatively. Each criterion is to be given one of the following assessments:

- Computational Assessment
    - High: computing time is much larger than that of other calculation steps; probably includes a simulation model, e.g. > 75% of the time for a fitness evaluation
    - Medium: moderate amount of computing time; may consist of a simulation model, e.g. 25-75% of the time for a fitness evaluation
    - Low: small amount of computing time, e.g. 5-25% of the time for a fitness evaluation
    - Trivial: trivial amount of computing time, e.g. < 5% of the time for a fitness evaluation
- Dimensionality Assessment
    - Trivial: e.g. a function of 1-3 variables
    - Low: e.g. a function of 4-10 variables
    - Medium: e.g. a function of 11-20 variables
    - High: e.g. a function of 21-49 variables
    - Very High: e.g. a function of 50+ variables
- Smoothness Assessment
    - Smooth: the function and its derivative are both continuous

- Non-smooth: one discontinuity present in either the function or its derivative
- Very non-smooth: two or more discontinuities present

In line with the two principles mentioned above, an ideal metamodel scope would include calculation steps that score (1) high in computational assessment, (2) as low as possible in dimensionality assessment, and (3) as smooth as possible in smoothness assessment. Based on these ideals, for each criterion, a subset of the calculation steps for which the metamodel could be a surrogate is selected. These are S_1, the set of calculation steps that are included in the metamodel scope when evaluating the steps against the computational intensity criterion; S_2, the best metamodel scope in terms of dimensionality; and S_3, the best metamodel scope in terms of smoothness.

The best metamodel scope is then defined as the set of calculation steps common to all three potential metamodel scopes. This is expressed mathematically by Eq. 5.6.

$$S_{Best} = \bigcap_{i=1}^{3} S_i \tag{5.6}$$

If $S_{Best} = \{0\}$, there is no global best definition for all three criteria. In this case, consider metamodel scopes that are good in two criteria, i.e. those defined by Eqs 5.7-5.9. The performance of each metamodel should be assessed in terms of fidelity (step 6 from Figure 5.1), and if there is no clear best performing metamodel, they should be assessed in terms of speed-up during optimization (step 7 from Section 5.2).

$$S_{1,2} = S_1 \bigcap S_2 \tag{5.7}$$

$$S_{1,3} = S_1 \bigcap S_3 \tag{5.8}$$

$$S_{2,3} = S_2 \bigcap S_3 \tag{5.9}$$

Similarly if $S_{1,2} = S_{1,3} = S_{2,3} = \{0\}$, consider the best metamodel according to each separate criterion and assess them according to fidelity and speed-up.

This systematic approach is demonstrated by application to two mathematical functions in Appendix A.

5.2.2 Post-Optimization Solution Checking (Step 8)

In order to check the accuracy of the results obtained using the metamodel-based optimisation, Broad et al. (2005) proposed an algorithm that checks certain solutions using the original fitness function that uses the simulation model. This is based on the assumption that the metamodels only provide an imperfect approximation, and so, the EA using a metamodel as a surrogate for the simulation model will converge to a solution near, but not equal to the global optimum, and hence some solutions need to be checked with the simulation model. The algorithm for achieiving this comprises the following steps:

1. Cache each new best solution found by the metamodel-based EA and evaluate it with the simulation model;
2. Track the top 40 solutions found by the EA and evaluate them with the simulation model; then

3. Conduct a local search until complete convergence occurs, using the best solution found by steps 1 and 2 as a starting point.

However, this approach becomes computationally intractable when the use of computationally intensive simulation models is required, as is the case for real case studies. Consequently, in such situations, the approach of Broad et al. (2005) needs to be modified, as discussed below. Assume T_{SIM} hours of CPU time is available for post-EA fitness evaluations with the simulation model, and knowing the average run-time of a simulation (t_{MODEL}), and the number of MCS simulations required for each fitness evaluation (N_{MCS}), the number of available fitness evaluations, N_{SIM}, can be calculated by Eq. 5.10. The question then becomes what is the best way to spend these evaluations? The number of simulations may be expressed as shown in Eq. 5.11.

$$N_{SIM} = \frac{T_{SIM}/t_{MODEL}}{N_{MCS}} \tag{5.10}$$

$$N_{SIM} = N_{NB} + N_{TOP} + N_{LS} \tag{5.11}$$

Where N_{NB} is the number of new best solutions evaluated with the simulation model; N_{TOP} is the number of top solutions that are tracked and evaluated with the simulation model; and N_{LS} is the number of local search iterations.

Considering the proposed limit of evaluations at each step of the algorithm of Broad et al. (2005), there is a need to modify the algorithm as follows.

Step1: Cache each new best solution found by the metamodel-based EA and evaluate every *n*-th solution with the simulation model, where $n = N_{NB\text{-}TOTAL}/N_{NB}$, and $N_{NB\text{-}TOTAL}$ is the total number of new best solutions the EA finds. The reasoning behind using this approach becomes clearer when one considers the expected metamodel performance as the EA proceeds. Initially, the metamodel will correctly identify a high

proportion of solutions as feasible or infeasible; then, near convergence, due to slight errors between the metamodel and simulation model, a higher proportion of solutions will be categorized incorrectly. However, the point during the EA run when this occurs is not known *a priori*. Therefore, the most robust way to ensure the likelihood that a very fit feasible solution is found from amongst the $N_{NB\text{-}TOTAL}$ solutions is to use an equi-spaced solution checking process in terms of order in which solutions are found.

Step 2: Track the top N_{TOP} solutions found by the EA and evaluate them after the EA has converged with the simulation model; then

Step 3: Conduct a local search for N_{LS} iterations using the best solution found by steps 1 and 2 as a starting point. Given the limited number of iterations, some consideration should be given to which local search algorithm to use (see Broad et al. (2006) for an evaluation of different local search methods designed specifically for post metamodel-based EA optimization).

5.3 Application to Risk-Based Optimal Design of WDSs

5.3.1 Background

Research into the optimal design of WDSs increased rapidly with the first application of EAs to the problem (Simpson et al. 1994). Most subsequent research focused solely on considering hydraulic criteria. The inclusion of water quality (e.g. chlorine dosing and tracking residuals) increased run-times compared with hydraulics-only optimization, as extended period simulations and shorter modelling time-steps were needed. Consequently, there are very few examples of this in the

literature (Broad et al. 2005; Hewitson and Dandy 2000), even though it is an important aspect to consider when designing real WDSs. There have been attempts to speed up the simulation of chlorine in WDSs (Constans et al. 2003), however, these have been limited to first-order decay models and a more generic approach to increasing computational efficiency is required. Broad et al. (2005) demonstrated the benefit that metamodels provide with regard to addressing this problem.

Computational issues associated with EA-based optimization are magnified significantly when considering uncertainty, as discussed in Section 5.1.1. Uncertainty specific to WDS and how it should be accounted for is detailed in the following sections.

Uncertainty

Data uncertainty has long been considered as another important factor to account for when designing a water distribution system (Lansey and Mays 1989; Su et al. 1987). Sources of uncertainty include future demands, pipe roughness (Bao and Mays 1989; Tolson et al. 2001; Tolson et al. 2004), chlorine decay rate (Tyagi 2003), and pipe bursts (Su et al. 1987). Each of these uncertainty sources can be categorized as "data uncertainty" (see Section 1.1), as reasonable estimates can be made to quantify the magnitude of uncertainty. The range of possible future demand values can be estimated based on a city's projected population growth. Existing pipe roughness variation can be quantified by inspecting the roughness at a number of locations across the network. Future pipe roughness can be estimated based on expected water quality, pipe materials and the system's design life. Chlorine decay rate variation can be estimated by considering the source water's natural variability in quality.

Risk Metrics

To account for this uncertainty, various definitions of risk metrics have been provided (Babayan et al. 2005; Cullinane et al. 1992; Duan et al. 1990; Farmani et al. 2005; Gargano and Pianese 2000; Kapelan et al. 2005; Khomsi et al. 1996; Ormsbee and Kessler 1990; Shinstine et al. 2002; Su et al. 1987; Xu and Goulter 1999). Gargano and Pianese (2000) considered a combined hydraulic (demand uncertainty) and mechanical (pipe burst) reliability metric and concluded that the overall contribution to reliability added by mechanical reliability is insignificant compared to hydraulic reliability. Therefore, mechanical reliability has not been considered in this research.

As mentioned earlier, reliability and vulnerability are two key risk metrics used in WRM optimization. Hashimoto et al. (1982) define these mathematically by the following equations (Eq. 5.12-5.13).

$$R = 1 - p_f = 1 - Pr\{f(\boldsymbol{x}) < \bar{y}\} = 1 - \lim_{n\to\infty}\left(\frac{1}{n}\sum_{k=1}^{n} w_k\right) \tag{5.12}$$

$$V = E[f(\boldsymbol{x})|f(\boldsymbol{x}) < \bar{y}] = \lim_{n\to\infty}\frac{1}{n}\sum_{\substack{k=1 \\ f(x_k)<\bar{y}}}^{n} e_k \tag{5.13}$$

Where R is reliability; p_f is the probability of failure; $f(\boldsymbol{x})$ is the joint probability distribution function of the vector of independent variables, $\boldsymbol{x}$; $\bar{y}$ is the failure threshold of the dependent variable, y; w is the failure indicator, as shown in Eq. 5.14; e is the expected value of the failure indicator, as shown in Eq. 5.15, and k is the MCS sample number.

$$w_k = \begin{cases} 1 & if\ f(x_k) < \bar{y} \\ 0 & otherwise \end{cases} \tag{5.14}$$

$$e_k = \begin{cases} \bar{y} - f(x_k) & if\ f(x_k) < \bar{y} \\ 0 & otherwise \end{cases} \tag{5.15}$$

For the optimal design of WDSs, both metrics may have hydraulic and water quality definitions. Hydraulic reliability is defined as the probability that pressure is above some prescribed minimum. Hydraulic vulnerability is defined as the expected pressure deficit when pressure is inadequate. Water quality reliability is defined as the probability that the residual chlorine concentration is above some minimum, which is often legislated for health reasons. Water quality vulnerability is defined as the expected minimum chlorine residual violation, should violation occur.

These metrics must initially be calculated at each node and, in turn, aggregated across the network. Two common methods to do this are "worst case" and "demand-weighted", and are defined in the generic case by Eqs. 5.16-5.19. Specific hydraulic and water quality equations can be identified trivially.

$$R_{WC} = \min_{j \in Nc}(R_j) \tag{5.16}$$

$$R_{DW} = \frac{\sum_{j \in n} DM_j R_j}{\sum_{j \in n} DM_j} \tag{5.17}$$

$$V_{WC} = \max_{j \in Nc}(V_j) \tag{5.18}$$

$$V_{DW} = \frac{\sum_{j \in n} DM_j V_j}{\sum_{j \in n} DM_j} \tag{5.19}$$

Where *j* is the node index, *n* is the number of nodes, *Nc* is the number of critical nodes.

The "worst case" method appears to be the most appropriate, as it ensures that all of a water utility's customers receive adequate supply. One negative aspect of

the "demand-weighted" method is that poor performance in a small area of the system could be masked by good performance in the majority of the system, especially for larger systems (i.e. real-world case studies).

Acceptable reliability (R^*) and vulnerability (V^*) values need to be identified for use in constraints that will guide the optimization. For example, these might be obtained from a water utility's contract with its regulator or by Key Performance Indicators set by the utility itself. Alternatively, risk-metrics could be included in a multi-objective optimization (Halhal et al. 1997; Kapelan et al. 2005), although this is beyond the scope of this paper. Generic reliability and vulnerability violation equations are given by Eqs. 5.20 and 5.21, respectively.

$$R_{Viol} = \max[0, R^* - R_{WC}] \tag{5.20}$$

$$V_{Viol} = \max[0, V_{WC} - V^*] \tag{5.21}$$

Risk metrics are often calculated using Monte Carlo Simulation (MCS). This research utilizes Hammersley Sampling (HS) due to its superior convergence and usefulness over other techniques (Kalagnanam and Diwekar 1997; Simpson et al. 2001). See Appendix B for more information.

Summary

To date, the authors are unaware of any literature where data uncertainty for demand, pipe roughness and decay rate has been considered in the EA-based optimization of a real-world problem. This is because the excessive run-times render this computationally impossible in a reasonable amount of time. Therefore, that is the problem that has been selected for testing the proposed process. Through the use of metamodelling and the formalized scope definition process presented in this paper,

this is able to be achieved for the first time, as demonstrated in the following sections of this paper.

Generically, the problem may be formulated by Eqs 5.22-5.27, where the objective function is Eq. 5.22, the decision variables are Eq 5.23, and the constraints are Eqs. 5.24-5.27.

$$\min z = f(\boldsymbol{x}) = \sum_{i=1}^{NP} UC_i L_i + NPV\left(\sum_{j=1}^{ND} C0_j\, Q_j T\right) \tag{5.22}$$

Where *UC* is the pipe unit cost (material and labour) (generally represented as a lookup table), *L* is the pipe length, *NP* is the number of pipe decision variables, *C0* is the chlorine dosing rate, *Q* is the average flow at the chlorine dosing point, *T* is the total time the chlorinator is dosing in a year, *ND* is the number of dosing points, and NPV is the net present value. The decision variables, **x**, are the pipe diameters and chlorine dosing rates, expressed mathematically in Eq. 5.23.

$$\boldsymbol{x} \equiv \boldsymbol{D} \bigcup \boldsymbol{C0} \tag{5.23}$$

$$R^*_{HYD} \leq R_{WC-HYD} \tag{5.24}$$

$$V_{WC-HYD} \leq V^*_{HYD} \tag{5.25}$$

$$R^*_{WQ} \leq R_{WC-WQ} \tag{5.26}$$

$$V_{WC-WQ} \leq V^*_{WQ} \tag{5.27}$$

Constraints in WDS optimization are commonly handled by penalty functions, hence the overall problem can be formulated as a single objective function, as given by Eq. 5.28.

$$\begin{aligned}\min z = f(\boldsymbol{x}) = & \sum_{i=1}^{NP} UC_i L_i + NPV\left(\sum_{j=1}^{ND} C0_j\, Q_j T\right) \\ & + PM_{R-HYD}\max[0, R^*_{HYD} - R_{WC-HYD}] \\ & + PM_{V-HYD}\max[0, V_{WC-HYD} - V^*_{HYD}] \\ & + PM_{R-WQ}\max\left[0, R^*_{WQ} - R_{WC-WQ}\right] \\ & + PM_{V-WQ}\max\left[0, V_{WC-WQ} - V^*_{WQ}\right]\end{aligned} \tag{5.28}$$

Where $PM_{R\text{-}HYD}$ and $PM_{R\text{-}WQ}$ are the hydraulic and water quality reliability penalty multipliers, respectively; and $PM_{V\text{-}HYD}$ and $PM_{V\text{-}WQ}$ are the hydraulic and water quality vulnerability penalty multipliers, respectively.

Clearly, there are many intermediate steps in computing this function. Hence the systematic metamodel scope definition process defined in Section 5.2.1 can be used to determine the best metamodel for optimization. This is done in Section 5.3.2.

5.3.2 Metamodel Scope Definition

Consider the metamodel scope definition selection process introduced in Section 2 applied to the single objective, risk-based optimal design of water distribution systems with hydraulic and water quality criteria considered, as outlined in Section 3.1. Figure 4 shows the calculation steps involved in calculating the fitness function, which is fairly complex when broken down into each calculation step. DM_{MULT}, HW_{MULT} and k are stochastically sampled variables representing a nodal demand multiplier, Hazen-Williams C coefficient multiplier and chlorine decay rate, respectively.

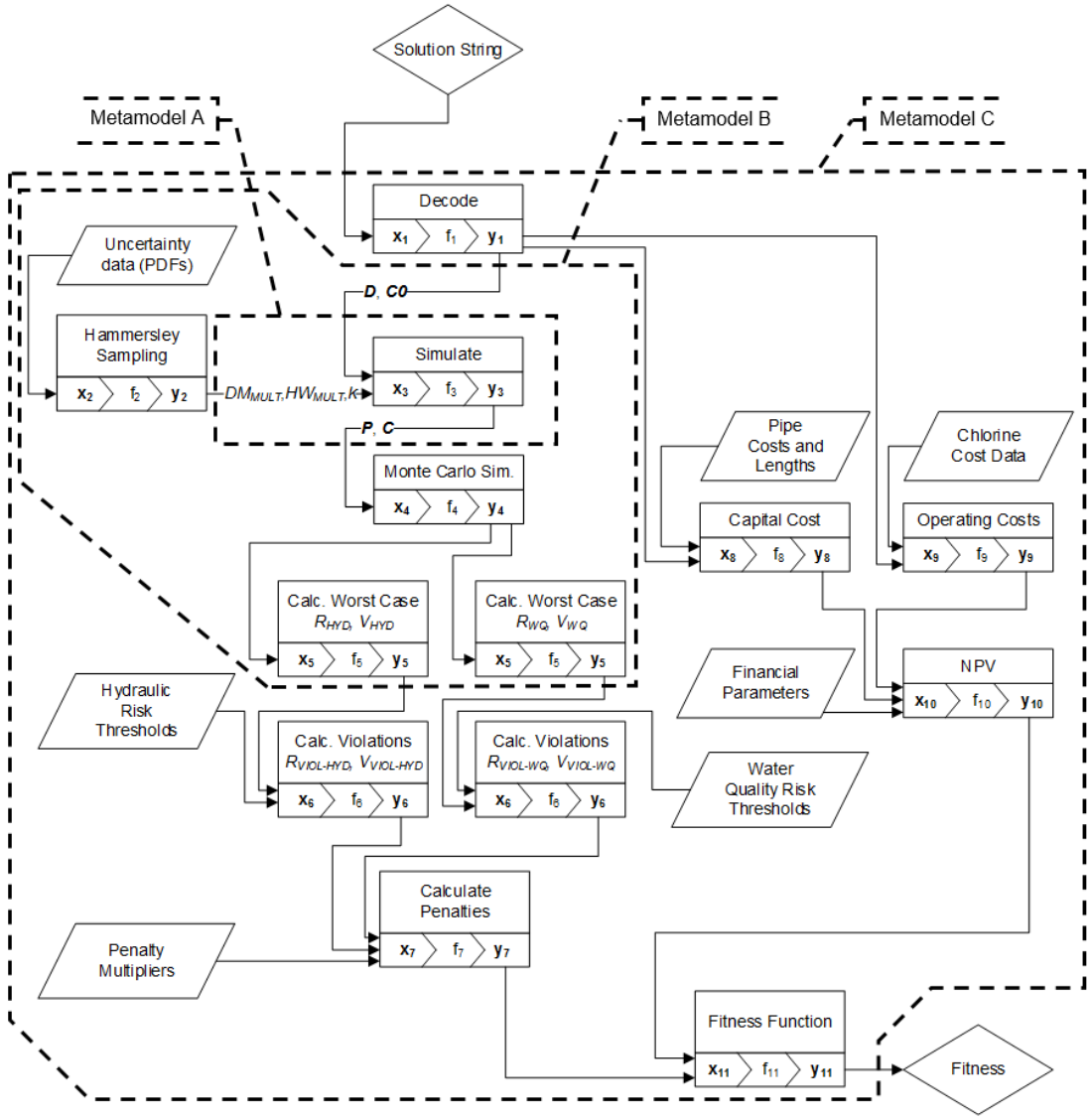


Figure 5-4. Fitness evaluation for risk-based optimization of WDS with metamodel scope options.

Table 5.1 provides a summary analysis of each calculation step, while extra detail on each step is provided below:

Step 1: Variable decoding will contain discontinuities for decisions that have discrete options (e.g. pipe decisions), but would be smooth for continuous decision variables (e.g. tank diameter). Hence the specific model needs to be considered.

Table 5.1. Assessment of fitness calculation steps for single-objective risk-based optimal design of water distribution systems.

	Calculation	Computational Assessment	Dimensionality Assessment[a]	Smoothness Assessment
1	Decode Variables	Trivial	Problem Dependent	Smooth-Non-smooth
2	Sample from PDFs	Low	Low-High	Smooth
3	Simulate Model	High	Model Dependent	Mostly smooth
4	Monte Carlo Simulation	Medium	See step 2	Very non-smooth
5	Worst Node Risk Values	Trivial	Model Dependent	Non-smooth
6	Risk Violations	Trivial	Low	Non-smooth
7	Risk Penalties	Trivial	Low	Smooth
8	Capital Cost	Trivial	Problem Dependent	Problem Dependent
9	Operating Cost	Trivial	Problem Dependent	Smooth
10	Net Present Value	Trivial	Low	Smooth
11	Objective Value	Trivial	N/A	Smooth

[a]"Problem Dependent" is a comment on the optimization formulation, whereas "Model Dependent" is a comment on the physical size of the simulation model. For example, a large model may have a simple or complex formulation, depending on the number of decision variables

Step 2: Sampling from random variables, such as future demands, pipe roughness and chlorine decay rate can vary significantly in dimensionality, depending upon the approach used. If the random variables are perfectly spatially correlated, then a single parameter can be used, e.g. future demand multiplier, which scales up/down all demand nodes in the network. However, if the spatial correlation is not very strong, then a variable for each location would be required. Hence, the number of variables is likely to increase significantly for larger models. For pipe roughness, the spatial correlation should be quite high for pipes in a similar category (e.g. concrete,

20-30 years old). Therefore, it is possible to consider a small number of categories of pipe, have a roughness multiplier for each category and hence keep the number of random variables small. Considering the equations governing HS (see Eqs. B.1-B.2 in Appendix B), which contain ceiling and floor functions, the smoothness will be very low. This should be expected, given HS is used in lieu of a pseudo-random number generator (random sampling, LHS).

Step 3: Modelling water quality is very computationally intensive, as discussed in Section 5.3.1. In relation to smoothness, model simulation should be fairly smooth for most decision variables. For example, larger pipes result in less head-loss and hence higher downstream pressures; decreased demand results in lower velocity and higher water age, and hence lower residual chlorine concentrations at the extremities of the network. Discontinuities would be present if the model included system controls.

Step 4: The Monte Carlo Simulation step to calculate reliability and vulnerability are non-smooth functions. Reliability is constrained $\in [0,1]$, while vulnerability is constrained ≥ 0. Hence the derivative will be discontinuous in both cases. The computational intensity of MCS is primarily dependent on the number of samples required for convergence, and secondarily on the number of random variables used. As discussed in Appendix B, Hammersley sampling can keep the number of MCS samples to a minimum. Nodal risk metrics need to be calculated at all critical nodes. For the purpose of metamodelling, a critical demand node is one that, for at least one sample of the generated calibration data, is the worst-performing node. It is important that all critical demand nodes are identified to ensure the optimization algorithm converges to a feasible solution (according to the metamodel).

At the same time, it is desirable to minimize the number of critical demand nodes and thus reduce the dimensionality of the metamodel. Broad et al. (2010) presented a quantitative method to determine the critical demand nodes for WDS optimization.

Step 5: The worst-case risk calculations will contain discontinuities in the derivative when considered as a function of the decision variables. There may be one part of the search-space where one node is the worst and another part where a different node is the worst. In-between, there may be a discontinuity where it jumps from one to the other. Dimensionality will depend on the number of critical nodes, and hence is model dependent.

Step 6: Risk violation calculations are trivial in terms of computational intensity, but are not smooth due to them being bounded (≥ 0), as demonstrated in Eqs. 5.33 and 5.34.

Step 7: Risk penalty calculations are trivial, as they each only consist of multiplication by a scalar.

Step 8: Capital cost calculations are simple, however, there may be some discontinuities when considering how unit costs vary with pipe diameter. These are determined by the market and there may be discontinuities in the price structure, as the preferred (cheaper) material (and hence roughness coefficient) may change as diameter increases. Also, labour costs may increase sharply as the preferred installation method may change as diameters increase (e.g. due to greater safety precautions needed).

Step 9: Operating cost calculations include simple scalar multiplication and the number of chlorinators will only be low (compared to other variables). Consequently, dimensionality is low.

Step 10: The NPV function is trivial and includes very few variables. Consequently, dimensionality is low.

Step 11: The objective function is then simply the sum of the NPV and penalty costs.

Based on the computational assessment outlined above, the metamodel should definitely include step 4 and possibly include step 5. Based on the dimensionality assessment, the metamodel should avoid steps 2 and 5 if random variables are not perfectly spatially distributed. Based on the smoothness assessment, the metamodel should avoid steps 5-7. Therefore, the best metamodel scope is for it to include calculation step 3 only (see Eq. 5.29, and "Metamodel A" in Figure 5.4). The input variables are the decision variables and the random variables are given by Eq. 5.30, and the output variables are the pressures and chlorine residuals at the critical nodes (Eq. 5.31). Initially the values at each node need to be calculated. Subsequently, the critical nodes can be determined using the method of Broad et al. (2010).

$$S_{Best} = \{f_3\} \tag{5.45}$$

$$\boldsymbol{x_{MM}} = \{\boldsymbol{dv}\} \bigcup DM_{MULT} \bigcup HW_{MULT} \bigcup k \tag{5.46}$$

$$\boldsymbol{y_{MM}} = \{P_i, \forall i \in N_{C-HYD}\} \bigcup \{C_j, \forall j \in N_{C-WQ}\} \tag{5.47}$$

Where DM_{MULT} is the demand multiplier that is applied to each node, HW_{MULT} is the pipe roughness coefficient multiplier that is applied to each pipe, k is the chlorine decay rate, P is the minimum nodal pressure over the EPS, C is the minimum nodal residual chlorine over the EPS, $N_{C\text{-}HYD}$ is the number of critical nodes for hydraulics, and $N_{C\text{-}WQ}$ is the number of critical nodes for water quality.

By way of comparison, and to demonstrate the benefit of undertaking a systematic approach to determining metamodel scope, two alternative scopes are presented here. One such scope is a metamodel that approximates the constrained variables, labelled "Metamodel B" in Figure 4. Another scope is a metamodel that approximates the entire fitness function, labelled "Metamodel C" in Figure 4. Both of these options for metamodel scope have been used by previous researchers in a range of applications (see Section 1.2), however, by using a systematic approach to determining metamodel scope, the problems with these alternative scopes are clearly identified. Metamodel B includes some non-smooth calculation steps but these are not high in computational intensity. Metamodel C maximizes potential time-saving by including all calculation steps, but any additional benefit this gives over Metamodel A would be significantly outweighed by the additional non-smooth functions it needs to approximate.

5.3.3 Case Studies

Following the determination of the best metamodel scope for the risk-based optimization of WDSs with EAs, the main metamodelling approach (steps 3-8 of section 5.2) is applied to two case studies to ensure performance is adequate in terms of fidelity to the simulation model, the ability to find optimal solutions, and computational speed-up. A simple case study (modified version of the New York Tunnels (NYT) problem) was used first, as this enables comparisons to be made between metamodel and non-metamodel approaches. The second case study, Pacific City, provides a real-world test of the proposed approach.

Artificial Neural Networks (ANNs) were selected as the metamodel type due to their demonstrated ability to model water quality variables in distribution systems

(May et al. 2008; Bowden et al. 2006; Gibbs et al. 2006). Multi-Layer Perceptrons (MLPs) were used as the specific type of ANN as they have been successfully used for a range of water quality models (Wu et al. 2014) and have shown to outperform other ANN types to forecast chlorine residuals in WDS (Gibbs et al. 2006).

Customised code was written to carry out all analyses presented in the following sections. It was written in C++ and compiled with Microsoft Visual Studio's compiler as a 32-bit executable. All runs were carried out on an Intel Core i7 hyper-threaded quad-core 870 @ 2.93 GHz CPU running 8GB RAM and Windows 7. Some steps of the metamodel development were split into separate batch processes to reduce wall-time (i.e. generating calibration data, training individual ANNs); however, only CPU times are presented here.

In a commercial setting, it is desirable to conduct optimization runs overnight, where the results from one run are reviewed during business hours and the problem formulation and/or optimization parameters are modified for a subsequent run the following night (Murphy, 2014). Hence, a value of 15 hours of CPU time has been assumed for T_{SIM}.

Razavi et al (2012b) recommend that the analyst time be taken into consideration also. For the appropriate analyst (i.e. an engineer who was familiar with GAs, ANNs and using them in metamodelling applications) that time would be trivial. The time required to develop the code to carry out these analyses should not be considered. If customized code were written for every metamodelling application, metamodelling would likely never become a worthwhile venture. It is the opinion of the authors that all researchers in the field of metamodel-enabled optimization of WDSs are working towards establishing that the technology is viable and that a

generic methodology can be established. If, and when, the research matures to that point robust software will be developed and used by practitioners that will be used as commonly as hydraulic modelling packages.

Optimization runs for nine scenarios were carried out. Scenario A, MM-EA Only, considered EA-based optimization with the metamodel only, as shown in Table 5.2. Scenarios B-H began with Scenario A, but then included various combinations of the solution checking parameters mentioned in Section 5.2.2 (see the following sections for the case study-specific values). Scenario I considered EA-based optimization with the simulation model only. Scenario I was only run for the NYT problem due to the long run-times for the real-world case study. Optimization runs for each scenario for each case study were repeated 30 times with different starting seeds to allow for the stochastic nature of EAs. It may be worthwhile for future research to consider different randomly seeded sets of calibration data and initial weights in the ANN metamodels due to their stochastic nature to see if there is a significant impact on the results.

All optimization runs were carried out using a Genetic Algorithm (GA), due to its proven performance for WDS optimization (Simpson et al. 1994; Savic and Walters 1997), recognizing that there is no single best algorithm for all WRM applications and that more research is required in characterizing fitness landscapes to assist in algorithm selection (Maier et al. 2014).

Table 5.2. Scenarios used in the two case studies.

Scenario	Model used in EA	Solution Checking	Case Studies
A	Metamodel	None	Both
B	Metamodel	Equal across all 3 methods	Both
C	Metamodel	Mostly new best and local search	Both
D	Metamodel	Mostly new best	Both
E	Metamodel	Mostly new best and top few	Both
F	Metamodel	Mostly top few and local search	Both
G	Metamodel	Mostly top few	Both
H	Metamodel	Mostly local search	Both
I	Simulation	None	1 only

New York Tunnels

The New York Tunnels (NYT) problem (originally presented by Schaake and Lai (1969) and first optimized using GAs by Dandy et al. (1996)) is used as a case study here for proof of concept purposes. It is a small problem and therefore a comparison can be made between optimization with the use of metamodels and a "traditional" (non-metamodel) approach (Scenario I). See Appendix C.1 for details of NYT and the modifications made to it for this work.

Metamodel development and performance

During the development of the metamodel, the parameters used by Broad et al. (2005) were used as a starting point. Some fine-tuning of the parameters by trial-and-error then yielded the final parameters used in the analysis, which are presented in Table 5.3.

Table 5.3. Metamodelling parameters used for NYT case study.

Parameter	Value
Metamodel type	Multi-layer perceptron (MLP)
Learning algorithm	Back-propagation
Total calibration data	10,000
Training data (% of total)	60%
Testing data (% of total)	20%
Validation data (% of total)	20%
Scaling bounds	$\in$[0.1, 0.9]
Initial weights	$\in$[-0.225, 0.225]
Epoch size	1
Learning rate	0.3
Momentum rate	0.5
Hidden layers	1
Hidden nodes	40

Calibration data were separated into training, testing and validation randomly, with the exception that solutions corresponding to the minima and maxima of each of the input and output variables were placed in the training set. This ensured that the training data covered the broadest possible range and that the metamodel was interpolating and not extrapolating.

Following the procedure for determining critical nodes outlined in Broad et al. (2010), it was discovered that there were three critical hydraulic nodes and two critical water quality nodes. The overall structure of the metamodel is presented in Table 5.4. This is consistent with Broad et al. (2005), who found that single MLPs per output was the best metamodel configuration in terms of fidelity to the surrogate model.

Table 5.4. Input and output variables for the various MLPs within the metamodel for New York Tunnels.

Multi-Layer Perceptron	Input Variables		Output Variable
	Decision Variables	Random Variables	
MLP-1	21 Pipe diameters	1 Demand multiplier 1 Roughness multiplier	Pressure @ Node 16
MLP-2	21 Pipe diameters	1 Demand multiplier 1 Roughness multiplier	Pressure @ Node 19
MLP-3	21 Pipe diameters	1 Demand multiplier 1 Roughness multiplier	Pressure @ Node 20
MLP-4	21 Pipe diameters 1 Chlorine dosing rate	1 Demand multiplier 1 Roughness multiplier 1 Decay rate	Residual chlorine @ Node 17
MLP-5	21 Pipe diameters 1 Chlorine dosing rate	1 Demand multiplier 1 Roughness multiplier 1 Decay rate	Residual chlorine @ Node 20

Table 5.5. Metamodel development results for NYT, showing the RMSE and R^2 for the validation set.

Criticality Type	Critical Node	RMSE (m or mg/L)	R^2
Hydraulic	16	0.12	0.9955
	19	0.20	0.9989
	20	0.12	0.9957
Water Quality	17	0.011	0.9995
	20	0.021	0.9989

The metamodel calibration results (root mean-squared error (RMSE) and coefficient of determination (R^2)) are presented in Table 5.5. Multiple metrics are used here for evaluation, as recommended by Bennett et al. (2013), as well as a visual representation of the worst performing MLP (MLP-1) in Figure 5. The results are for the validation set, i.e. an independent data set not used for calibration. The results demonstrate a high fidelity between the metamodel and the simulation model, which gives confidence in the metamodel's ability to be used for optimization.

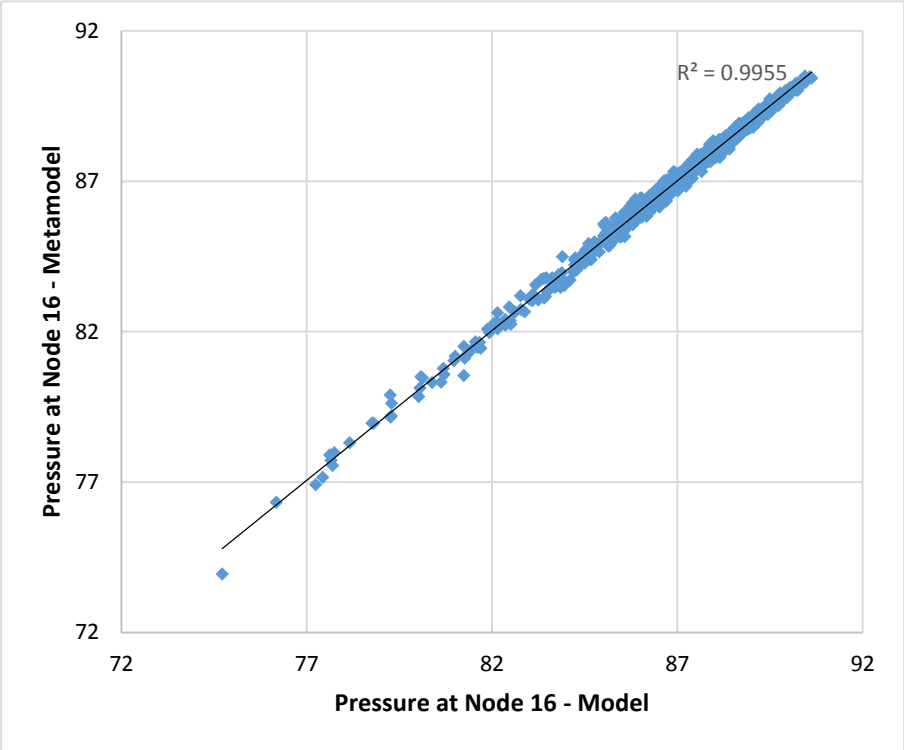


Figure 5-5. Comparison between simulation model and metamodel for critical hydraulic node, 16, for the New York Tunnels case study.

Optimization Approach

GA parameters were taken from Broad et al. (2005) (where the same case study was used) and modified by trial and error. The final parameter values are given in Table 5.6.

Table 5.6. Genetic Algorithm parameters used for NYT.

Parameter	Value
Coding	Integer
Population	200
Generations	1000
Crossover Points	1
Crossover Rate	0.8
Bit-wise Mutation Rate	0.02
Elitism	On
Hydraulic Reliability Penalty Rate	$\$10^9$
Hydraulic Vulnerability Penalty Rate	$\$10^9$/m
Water Quality Reliability Penalty Rate	$\$10^9$
Water Quality Vulnerability Penalty Rate	$\$10^{10}$/mg/L
Monte Carlo Samples per Evaluation	100

As discussed in Section 5.3.3, a value of 15 hours is assumed for T_{SIM}. Therefore, based on the average simulation time and N_{MCS} of 100, 10,588 solutions can be checked. These solutions are distributed differently for the seven different scenarios, detailed in Table 5.7.

Table 5.7. Number of fitness evaluations of each post-EA solution checking type for different scenarios for Pacific City.

Scenario	N_{NB}	N_{TOP}	N_{LS}	N_{SIM}
B	3,529	3,529	3,530	10,588
C	5,293	1	5,294	10,588
D	10,586	1	1	10,588
E	5,293	5,294	1	10,588
F	1	5,293	5,294	10,588
G	1	10,586	1	10,588
H	1	1	10,586	10,588

Optimization Results

Optimization results for thirty runs with different random number seeds are presented in Table 5.8. Scenario A frequently converged to a good solution ($24.0 million) that was seen to be feasible when checked by the simulation model. This was slightly higher (1.5%) than the best solution found without the metamodel (Scenario I, $23.6 million). The solution checking process provided minor benefits only. Considering the median values, there was no significant improvement between Scenario A and Scenarios B-H. However, the solution checking did enable the best overall solution to be found in one run out of thirty for the scenarios that included local search.

The relatively minor benefit provided by the post-EA solution checking algorithm seems to have primarily been due to the good performance of MM-EA, which is due to the high-fidelity metamodels that were calibrated.

Table 5.8. Optimization results for New York Tunnels. Statistics of NPV shown, as well as frequency that the best solution was found for 30 runs per scenario.

Scenario	Min	Mean	Median	Max	Std. Dev.	Best Frequency
A	2.40E+07	2.43E+07	2.40E+07	2.53E+07	5.17E+05	0/30
B	2.36E+07	2.40E+07	2.40E+07	2.40E+07	6.59E+04	1/30
C	2.36E+07	2.40E+07	2.40E+07	2.40E+07	6.59E+04	1/30
D	2.40E+07	2.40E+07	2.40E+07	2.40E+07	0	0/30
E	2.40E+07	2.40E+07	2.40E+07	2.40E+07	0	0/30
F	2.36E+07	2.40E+07	2.40E+07	2.40E+07	6.59E+04	1/30
G	2.40E+07	2.40E+07	2.40E+07	2.40E+07	0	0/30
H	2.36E+07	2.40E+07	2.40E+07	2.40E+07	6.59E+04	1/30
I	2.36E+07	2.40E+07	2.40E+07	2.40E+07	2.51E+05	2/30

Table 5.9 provides a summary of run-times for NYT. Considering only the EA, metamodelling provided a speed-up of 707 times (70,700%) over the non-metamodelling approach. Factoring in the extra overheads required for metamodel calibration and post-EA checking of some solutions, the speed-up factor was still 50 (5000%). While provision was made to allow for 15 hours of checking solutions with the simulation model, it was found that the solution checking algorithm converged early and that, on average, this step took only 2.7 hours.

Table 5.9. CPU times (hours) of each metamodelling step and comparison to non-metamodelling approach for NYT.

Metamodelling Step	With Metamodel	Without Metamodel
Generate training data	0.2	N/A
Determine critical nodes	0.1	N/A
Train ANNs	2.3	N/A
Run EA	0.4	283
Check solutions with simulation model	2.7	N/A
Total	5.7	283

Alternative metamodel scopes (approximating constrained variables and the entire fitness function) to that determined by the systematic approach presented in this paper were shown to be inferior (see Section 3.2, last paragraph). However, it is worth considering the impact on run-times of these alternatives. In both cases (Metamodel B and Metamodel C), running the EA would have been up to 100 times faster (as the Monte Carlo loop would be avoided). However, the time needed to generate training data would be 100 times longer, assuming the same number of training data are used due to the need for each data point to include a Monte Carlo loop. Overall, the total run-time would increase to approximately 25.1 hours. This is

significantly worse than if the systematic approach to determining scope were used (440% increase).

Pacific City

The following case study, named Pacific City, has been provided by Optimatics (2014) and is based on a commercial project undertaken for a client. Many details of the case study have been modified (e.g. name, co-ordinates, node elevation, pipe lengths, etc.) for security purposes and to maintain the client's anonymity (see Appendix C.2 for these details; the EPANET input file for this network is provided as supplementary material). However, in terms of model complexity, the problem still provides a real-world test case for risk-based optimization of WDSs using metamodels.

Metamodel development and performance

Three critical nodes were found for pressure while four were found for chlorine residual (see Figure C.1 in Appendix C.2 for their locations). For ease of reference, these are referred to as P1-P3 and C1-C4. The metamodel structure is given in Table 5.10.

The best metamodel calibration parameters were selected by trial-and-error. It was found that the same parameters as those used for the NYT problem performed best. Calibration results are presented in Table 5.11, most of which show very high fidelity to the original simulation model. Critical chlorine at node 4 had a lower coefficient of determination than that at all other nodes. However, the RMSE was still low and the comparison between model and metamodel shown in Figure 5.6 indicates sufficiently high fidelity. Therefore, optimisation runs could be carried out with confidence.

Table 5.10. Input and output variables for the various MLPs within the metamodel for New York Tunnels.

Multi-Layer Perceptron	Input Variables		Output Variable
	Decision Variables	Random Variables	
MLP-1	23 Pipe diameters	1 Demand multiplier 1 Roughness multiplier	Pressure @ Node P1
MLP-2	23 Pipe diameters	1 Demand multiplier 1 Roughness multiplier	Pressure @ Node P2
MLP-3	23 Pipe diameters	1 Demand multiplier 1 Roughness multiplier	Pressure @ Node P3
MLP-4	23 Pipe diameters 6 Chlorine dosing rates	1 Demand multiplier 1 Roughness multiplier 1 Decay rate	Residual chlorine @ Node C1
MLP-5	23 Pipe diameters 6 Chlorine dosing rates	1 Demand multiplier 1 Roughness multiplier 1 Decay rate	Residual chlorine @ Node C2
MLP-6	23 Pipe diameters 6 Chlorine dosing rates	1 Demand multiplier 1 Roughness multiplier 1 Decay rate	Residual chlorine @ Node C3
MLP-7	23 Pipe diameters 6 Chlorine dosing rates	1 Demand multiplier 1 Roughness multiplier 1 Decay rate	Residual chlorine @ Node C4

Table 5.11. Metamodel calibration results for Pacific City.

Critical Node	RMSE (m or mg/L)	R^2
P1	0.006	0.9998
P2	0.009	0.9998
P3	0.033	0.9986
C1	0.019	0.9985
C2	0.008	0.9998
C3	0.007	0.9998
C4	0.077	0.9727

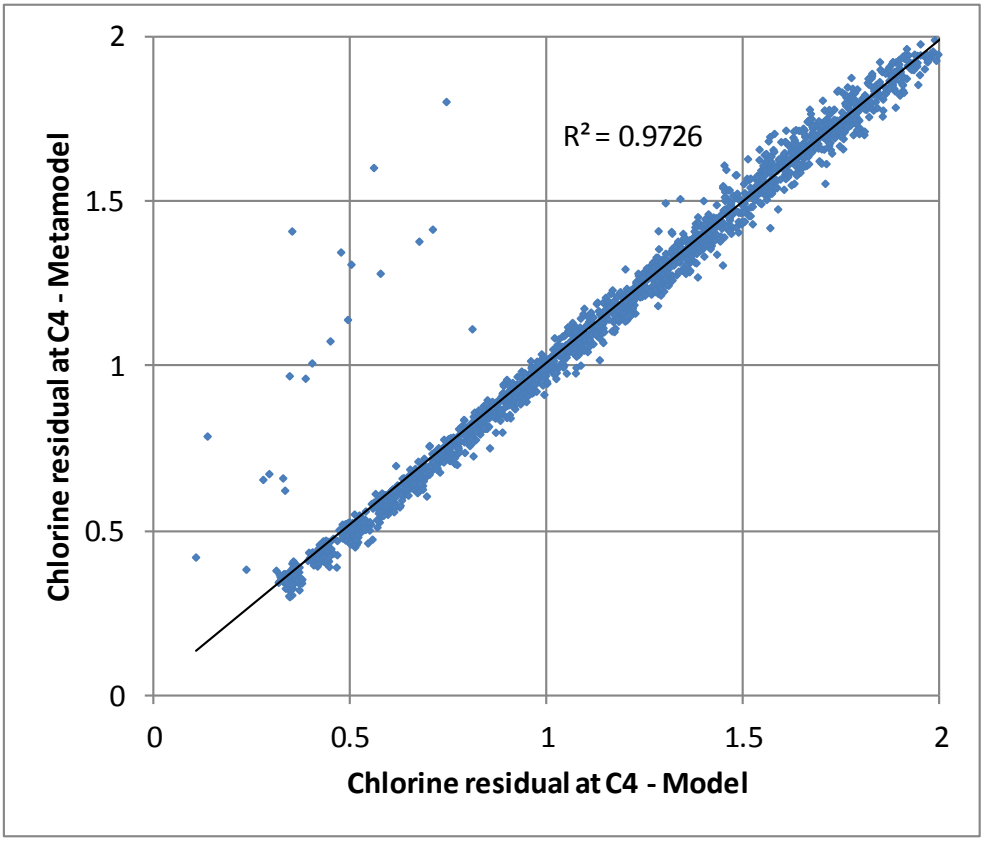


Figure 5-6. Comparison between simulation model and metamodel for critical chlorine node, C4, for the Pacific City case study.

Optimization Approach

The same optimization parameters were used for Pacific City as were used for NYT (see Table 5.6). In terms of solution checking, assuming T_{SIM} = 15 hours and N_{MCS} = 100, using the average simulation time there are only 28 fitness evaluations available. The distribution of these evaluations by each checking method for scenarios B-H are given in Table 5.12.

Table 5.12. Number of fitness evaluations of each post-EA solution checking type for different scenarios for Pacific City.

Scenario	N_{NB}	N_{TOP}	N_{LS}	N_{SIM}
B	9	9	10	28
C	13	1	14	28
D	26	1	1	28
E	13	14	1	28
F	1	13	14	28
G	1	26	1	28
H	1	1	26	28

Optimization Results

The best solutions from each of the thirty repeated runs for each scenario are presented in Table 5.13 (all solutions are feasible). Note the scenario A results are the solutions to which the metamodel-based EA converged. When all 30 of these solutions were checked with the simulation model, they were found to be infeasible.

Table 5.13. Optimization results for Pacific City. Statistics of NPV shown, as well as frequency with which the best solution was found for 30 runs per scenario.

Scenario	Min	Mean	Median	Max	Std. Dev.	Best Frequency
A	9.30E+06	9.33E+06	9.33E+06	9.36E+06	1.98E+04	0
B	8.86E+06	9.72E+06	9.95E+06	1.05E+07	6.48E+05	4/30
C	8.86E+06	9.88E+06	9.84E+06	1.10E+07	7.78E+05	4/30
D	9.49E+06	9.89E+06	9.78E+06	1.05E+07	3.58E+05	0
E	9.66E+06	1.04E+07	1.00E+07	1.22E+07	9.08E+05	0
F	1.25E+07	1.38E+07	1.38E+07	1.60E+07	1.37E+06	0
G	1.39E+07	1.49E+07	1.40E+07	1.65E+07	1.30E+06	0
H	1.24E+07	1.43E+07	1.40E+07	1.59E+07	1.34E+06	0

The first conclusion to draw from these results is that N_{NB} is a very important parameter, as indicated by the relatively good performance of scenarios B-E compared with that of scenarios F-H. Without adequate tracking of the progress of the EA, a good feasible solution cannot be found to seed the local search.

The two scenarios that found the overall best solution were B and C, each finding a solution with a cost of $8.86 million in four of the 30 runs. With scenario B having the best mean and better standard deviation compared to scenario C, scenario B is the recommended procedure to use, i.e. a fairly equal balance between the three solution checking strategies. This supports the reasoning given by Broad et al. (2005) for using each of these strategies.

Comparing the mean of scenario B with the mean of the EA only scenario shows a difference of 4%. This small difference indicates that although the EA converged to an infeasible solution for Scenario A, it was still in the "ballpark" compared with the overall best solution found.

Table 5.14 provides a summary of run-times for Pacific City. Due to the excessive run-time for the "without metamodel" scenario, only an estimate could be made based on average simulation time and EA and MCS parameters. Considering only the EA, metamodelling provided a speed-up of 171,000 times (1.7×10^7%) over the non-metamodelling approach. Factoring in the extra overheads required for metamodel calibration and post-EA checking of some solutions, the speed-up factor was still 1375 (137,500%). Thus, optimization is possible for this case study with metamodels, whereas using the traditional (non-metamodel) approach optimization is not possible in a reasonable amount of computer time.

Table 5.14. CPU times (hours) of each metamodelling step and comparison to non-metamodelling approach (estimated) for Pacific City.

Metamodelling Step	With Metamodel	Without Metamodel
Generate training data	43	N/A
Determine critical nodes	0.5	N/A
Train ANNs	3.2	N/A
Run EA	0.5	85,556
Check solutions with simulation model	15	N/A
Total	62.2	85,556

Repeating the analysis of run-times undertaken for New York Tunnels using the alternative metamodel scopes mentioned in Section 3.2, the overall run-time would increase to 4318 hours, which is completely impractical. As a percentage, this is an increase of 6943% (cf. 440% for New York Tunnels), indicating that the slower the model is (and the greater potential benefit that metamodelling could provide), the worse these alternative scopes are likely to be; again reinforcing the need to follow the systematic approach to determining metamodel scope.

5.4 Summary and Conclusions

This paper has made a number of significant contributions in relation to the use of meta-modelling for the speed-up of EAs, and hence the applicability of EAs to real-world problems, as suggested by Maier et al. (2014), as outlined below:

1. A formal, systematic approach for identifying which subset of the fitness function should be approximated by a metamodel so as to maximize the fidelity of the metamodel and achieve the greatest computational speed-up (termed

the metamodel scope) was introduced in this paper for the first time. The approach involves assessing each calculation step associated with the calculation of a fitness function against three key criteria: (1) computation time; (2) dimensionality; and (3) smoothness. A procedure is then followed to determine which of these calculation steps should be included within the metamodel scope based on these three criteria, where the use of metamodels is considered favourable if a calculation step exhibits the following properties: (1) high computation time; (2) low dimensionality; and/or (3) high smoothness.

2. A formal discussion of how this approach applies to the risk-based optimization of the design of water distribution systems was provided. As part of this discussion, all components of the fitness function used in this class of problem were identified and classified in accordance with the three criteria of the proposed approach (i.e. computation time, dimensionality, smoothness). Due to the generic nature of this discussion, the results of this assessment are transferrable to a large number of problems of this type.

3. The proposed process was applied to two case studies for the risk-based optimal design of water distribution systems using EAs, considering both hydraulic and water quality performance, which has not been done before. The second case study consists of a complex WDS based on a real network that has not been used previously in the literature. As the EPANET input file for this case study has been provided as supplementary material, this case study provides a useful, real-life benchmark that can be used by others, which is something that is needed in order to progress research in the field of the application of EAs to water resources problems (Maier et al., 2014). The benefits of the proposed approach compared with the

currently used ad-hoc meta-model building approach are illustrated by means of discussion. In addition, the overall benefits of the proposed approach in relation to its intended purpose of producing high-fidelity metamodels and speeding up EA optimization are demonstrated via the results of the two case studies. To the authors' knowledge, the second case study results are the first time this has been achieved for a real-world problem that accounts for data uncertainty in demand, pipe roughness and chlorine decay rate. In this case the optimization time was reduced from an impossibly long time (an estimated 85,000 hours of CPU time) to something reasonable (62 hours).

5.5 Acknowledgments

The research presented in this paper was funded by an Australian Postgraduate Award and the former Cooperative Research Centre for Water Quality and Treatment.

The authors would also like to thank Optimatics for providing details of the Pacific City case study.

Chapter 6

Conclusions and Recommendations

"And you should imitate me,

just as I (St. Paul) imitate Christ."

1 Corinthians 11:1 (NLT)

The optimization of WDS is a computationally intensive task when the factors considered in the formulation include those that a planning engineer typically needs to consider, including water quality and data uncertainty; and even more so, when a large, complex model is the subject of the optimization. Through the use of metamodels that act as fast-solving approximations to simulation models, computational intensity can be reduced to a practical level. This thesis demonstrates the effectiveness of using metamodelling and introduces guidelines for the development and use of metamodels for WDS optimization.

6.1 Research Contributions

There were two main aims of this research:

1. To incorporate three aspects of realism into the optimization of WDS, namely (1) water quality; (2) real-world systems; and (3) data uncertainty.
2. To develop a robust methodology for the use of metamodelling in WDS optimization.

Aim #1 was achieved progressively when considering the four publications in this thesis. Water quality criteria were included in each publication, real world systems were examined as case studies in the last two publications and data uncertainty was addressed in the last paper.

Aim #2 was also achieved progressively. A methodology was developed and proposed in the first publication. One step of the methodology, local search, was developed further in the second paper. Further modifications were made in the last two publications as real world systems were examined and data uncertainty considered. The final methodology is given in Figure 6.1.

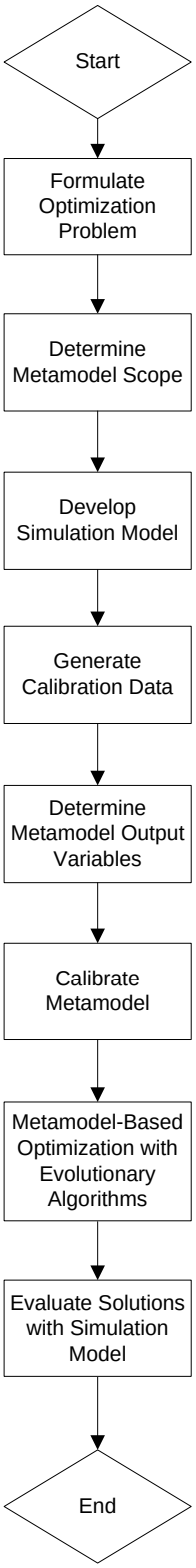


Figure 6-1 Methodology for the optimization of water distribution systems using metamodels

The key contributions to the development of a methodology for the use of metamodels to improve the computational efficiency of EA-based WDS optimization include the following:

- The development of a systematic approach to determine the most appropriate metamodel scope. The approach includes the deconstruction of the optimization problem's fitness function into a number of calculation steps. Each calculation step is assessed using several criteria to determine whether it should or should not be included among the calculation steps for which the metamodel acts as a surrogate. The methodology is generic to any metamodelling application and is designed to maximize fidelity to the original problem and to minimize the overall computational intensity.

- A methodology for determining metamodel output variables. The number of potential metamodel output variables is vast due to the geographical layout of the WDS (i.e. every node corresponds to another potential output variable). The developed methodology significantly reduces the number of output variables, which in turn reduces the computational time required for calibration. The methodology was developed specifically for WDS applications, however, it could be made more generic with some modifications.

- Demonstration that ANN metamodels are able to approximate a hydraulic simulation model. This research demonstrated that in four case study applications, ANN metamodels were able to act as surrogates for hydraulic models that ranged in size and complexity.

- An algorithm for checking solutions with the original simulation model. The algorithm, which is generic to any metamodelling application, was developed with the assumed principle that a metamodel will act as a high fidelity approximation to a simulation but will not be a *perfect* representation. Therefore, a metamodel coupled with an EA should be able to obtain a solution close to the true optimum. The solution checking algorithm was designed to find a solution as close as possible to the true optimum, as evaluated with the original simulation model.

- The methodology as a whole. Prior to this research the use of ANN metamodels to improve the computational efficiency of EA-based WDS optimization was very limited (Lingireddy and Ormsbee 1998). The methodology proposed in this thesis was developed with the aim to reduce computational intensity while still finding solutions that are as close to the global optimum as possible. The results from the four case studies demonstrate the effectiveness of the methodology as a whole.

6.2 Recommendations for Future Work

There are several possible research paths that can build upon the work presented in this thesis. These possibilities include:

1. The results presented in this research are all positive in that they demonstrate the benefit of using metamodels to reduce the computational intensity of WDS optimization. Four different case studies were examined in total, so a key requirement for future research is to apply the overall methodology (Figure 6.1) to several other case studies to confirm its

validity as a general approach. These case studies should include an increased number of decision variables and types of decisions.

2. It is imperative that all future research must examine complex hydraulic systems (e.g. Wallan, Pacific City, or larger in size). A key finding of this research is that the benefit of metamodels in terms of computational speed-up is greater for more complex systems and when more realistic factors are considered (e.g. data uncertainty). If future research only examines small WDS, there is a danger that the benefits of using metamodels will be so small that researchers conclude it is not worth using metamodels at all.

3. Other researchers who have used metamodels to speed-up EA-based optimization of WDS have periodically re-calibrated metamodels during the EA run (Behzadian et al. 2009, di Pierro et al. 2009, Bi and Dandy 2013). To date, as far as the writer is aware, there does not appear to be a comprehensive comparison study between that approach and the approach used in this research that tests complex simulation models and considers the run-time of all steps associated with developing and running metamodels. Such research would be beneficial to determine whether one approach was better than the other or whether the favourability of each approach was problem-dependent.

4. Estimate the accuracy of the simulation model and use it as a guide to determine the target accuracy of the metamodel. For example, there is little benefit in ensuring a metamodel for chlorine residuals is accurate to within 0.01 mg/L if the simulation model it is aiming to approximate is only accurate to within 0.1 mg/L. The advantage of this would be to reduce the

time required to calibrate the metamodel. Similarly, when using Monte Carlo simulation (MCS) to account for data uncertainty, there may be a way to determine the required accuracy of estimating risk-metrics and relating this to the number of MCS samples.

5. Consider using different ranges for decision variables when generating calibration data. This is based on the observation that ANN metamodels in particular perform better as interpolators than extrapolators. For example, consider an optimization formulation with a chlorine dosing range of [1.0, 3.0]. In this case, it may be beneficial to generate training data in the region [0.9, 3.1]. Also, consider a pipe decision with discrete options corresponding to commercially-available pipe sizes. While the EA must select one of these discrete options, there is no reason why the calibration data need to be discrete; there may be some benefit in generating calibration data in the continuous space. The advantage of carrying out this research might result in more accurate metamodels.

6. ANNs have often been used for modelling water resources in lieu of developing a process-based simulation model. This has been particularly effective in cases where the relationship between input and output variables is unknown, which is often the case with natural systems. In contrast, where ANNs are used as metamodels, the input/ouput relationship is known completely, as it is based on a simulation model. Therefore, there may be scope to somehow use this knowledge to modify how ANN metamodels are constructed. Future research may be able to leverage this known input/output relationship to (i) determine optimal

ANN architecture, or (ii) determine ANN weights (or initial weights). This could then result in reduced metamodel calibration time and/or improved metamodel fidelity to the simulation model.

7. The research presented in this thesis is specific to developing metamodels for optimizing the design and operations of WDSs. Useful future research would be to modify the methodology to make it more generic so as to be applicable to other water resources applications.

8. The results presented in the Case Studies throughout this thesis included carrying out repeated optimization runs with different random seeds. This is necessary to evaluate the robustness of the metamodels' ability to find optimal solutions when linked with EAs. The EAs are the main source of randomness in the entire metamodelling methodology, however, there are other sources such as the random generation of calibration data and initial random weights in the ANN metamodels. It would be useful for future research to use different random seeds and repeat the data generation and training steps several times, just like what is done with the EAs. This will ensure the robustness of the methodology is fully tested.

Bibliography

Ahuja, R. K., Ergun, O., Orlin, J. B., and Punnen, A. P. (2002). "A survey of very large-scale neighbourhood search techniques," *Discrete Applied Mathematics*, vol. 123, pp. 75-102.

Aly, A. H., and Peralta, R. C. (1999). "Optimal design of aquifer cleanup systems under uncertainty using a neural network and a genetic algorithm." *Water Resources Research*, 35(8), 2523-2532.

Babayan, A., Kapelan, Z., Savic, D., and Walters, G. (2005). "Least-cost design of water distribution networks under demand uncertainty." *Journal of Water Resources Planning and Management*, 131(5), 375-382.

Bao, Y., and Mays, L. W. (1989). "Model for Water Distribution System Reliability." *Journal of Hydraulic Engineering-ASCE*, 116(9), 1119-1137.

Bau, D. A., and Mayer, A. S. (2006). "Stochastic management of pump-and-treat strategies using surrogate functions." *Advances in water resources*, 29(12), 1901-1917.

Bauman, L. (2013). "New methods of uncertainty quantification for mixed discrete-continuous variable models." *Report No. SAND2013-5145*. Sandia National Laboratories (SNL-CA), Livermore, CA (United States).

Behzadian, K., Kapelan, Z., Savic, D., and Ardeshir, A. (2009). "Stochastic sampling design using a multi-objective genetic algorithm and adaptive neural networks." *Environmental Modelling & Software*, 24(4), 530-541.

Bennett, N. D., Croke, B. F., Guariso, G., Guillaume, J. H., Hamilton, S. H., Jakeman, A. J., Marsili-Libelli. S., Newham, L. T.H., Norton, J. P., Perrin, C., Pierce, S. A., Robson, B., Seppelt, R., Voinov, A. A., Fath, B. D., Andreassian, V. (2013). "Characterising performance of environmental models." *Environmental Modelling & Software*, *40*, 1-20.

Bi, W., and Dandy, G. C. (2013). "Optimization of Water Distribution Systems Using Online Retrained Metamodels." *Journal of Water Resources Planning and Management-ASCE*, 10.1061/(ASCE)WR.1943-5452.0000419.

Blanning, R. W. (1975). "The Construction and Implementation of Metamodels." *Simulation*, 24(6), 177-184.

Boccelli, D. L., Tryby, M. E., Uber, J. G., Rossman, L. A., Zierolf, M. L., and Polycarpou, M. M. (1998). "Optimal Scheduling of Booster Disinfection in Water Distribution Systems." *Journal of Water Resources Planning and Management - ASCE*, 124(2), 99-111.

Borgonovo, E., Castaings, W., and Tarantola, S. (2012). "Model emulation and moment-independent sensitivity analysis: An application to environmental modelling." *Environmental Modelling & Software*, 34, 105-115.

Bowden, G. J., Maier, H. R., and Dandy, G. C. (2002) "Optimal division of data for neural network models in water resources applications" *Water Resources Research*, 38(2), 1-11.

Bowden, G. J., Nixon, J. B., Dandy, G. C., Maier, H. R., and Holmes, M. (2006). "Forecasting chlorine residuals in a water distribution system using a general regression neural network." Mathematical and Computer Modelling, 44(5-6), 469-484.

Broad, D. R. (2014a). "EPANET Input File for New York Tunnels with Water Quality", PANGAEA Data Archiving & Publication, http://issues.pangaea.de/browse/PDI-7477

Broad, D. R. (2014b). "EPANET Input File for Pacific City", PANGAEA Data Archiving & Publication, http://issues.pangaea.de/browse/PDI-7478

Broad, D. R., Dandy, G. C., and Maier, H. R. (2005). "Water distribution system optimization using metamodels." *Journal of Water Resources Planning and Management-Asce*, 131(3), 172-180.

Broad, D. R., Dandy, G. C., Maier, H. R., and Nixon, J. B. (2006). "Improving Metamodel-based Optimization of Water Distribution Systems with Local Search." IEEE World Congress on Computational Intelligence, 16-21 July 2006, Vancouver, BC, Canada.

Broad, D. R., Maier, H. R., and Dandy, G. C. (2010). "Optimal Operation of Complex Water Distribution Systems Using Metamodels." *Journal of Water Resources Planning and Management*, 136(4), 433-443.

Caballero, J. A., & Grossmann, I. E. (2008). "An algorithm for the use of surrogate models in modular flowsheet optimization." *AIChE Journal*, *54*(10), 2633-2650.

Carnevale, C., Finzi, G., Guariso, G., Pisoni, E., & Volta, M. (2012). "Surrogate models to compute optimal air quality planning policies at a regional scale." *Environmental Modelling & Software*, *34*, 44-50.

Constans, S., Bremond, B., and Morel, P. (2003). "Simulation and Control of Chlorine Levels in Water Distribution Networks." *Journal of Water Resources Planning and Management - ASCE*, 129(2), 135-145.

Cullinane, M. J., Lansey, K. E., and Mays, L. W. (1992). "Optimization-Availability-Based Design of Water-Distribution Networks." *Journal of Hydraulic Engineering-ASCE*, 118(3), 420-441.

Cunha, M. C., and Sousa, J., (1999). "Water Distribution Network Design Optimization: Simulated Annealing Approach," *Journal of Water Resources Planning and Management-ASCE*, vol. 125, pp. 215-221.

Dandy, G. C., and Hewitson, C. (2000). "Optimising hydraulics and water quality in water distribution networks using genetic algorithms." *Proceedings, Joint Conference on Water Resources Engineering and Water Resources Planning and Management, ASCE (on CD-ROM), Minneapolis, Minnesota*.

Dandy, G. C., Simpson, A. R., and Murphy, L. J. (1996). "An improved genetic algorithm for pipe network optimization." *Water Resources Research*, 32(2), 449-458.

Deuerlein, J. W. (2008). "Decomposition Model of a General Water Supply Network Graph." *Journal of Hydraulic Engineering, - ASCE*, 134(6), 822-832.

di Pierro, F., Khu, S.T., Savic, D., Berardi, L., (2009). "Efficient multi-objective optimal design of water distribution networks on a budget of simulations using hybrid algorithms." *Environmental Modelling & Software*, 24 (2), 202-213.

Duan, N., Mays, L. W., and Lansey, K. E. (1990). "Optimal Reliability-Based Design of Pumping and Distribution Systems." *Journal of Hydraulic Engineering-ASCE*, 116(2), 249-268.

Espinoza, F.P. and Minsker, B.S., (2006), "Effects of Local Search Algorithms on Groundwater Remediation Optimization using a Self-Adaptive Hybrid Genetic Algorithm", *Journal of Computing in Civil Engineering*, 20(6), 420-430

Eusuff, M. M. and Lansey, K. E. (2003). "Optimization of Water Distribution Network Design Using the Shuffled Frog Leaping Algorithm", *Journal of Water Resources Planning and Management*, 129 (3), 210-225.

Farmani, R., Walters, G. A., and Savic, D. A. (2005). "Trade-off between Total Cost and Reliability for Anytown Water Distribution Network." *Journal of Water Resources Planning & Management*, 131(3), 161-171.

Gargano, R., and Pianese, D. (2000). "Reliability as Tool for Hydraulic Network Planning." *Journal of Hydraulic Engineering-ASCE*, 126(5), 354-364.

Geem, Z. W. (2006). "Optimal cost design of water distribution networks using harmony search." *Engineering Optimization*, 38 (3), 259-277.

Gessler, J. (1985). "Pipe network optimization by enumeration," Computer applications in water resources, New York, N.Y..

Gibbs, M. S., Maier, H. R., and Dandy, G. C. (2004). "Applying Fitness Landscape Measures to Water Distribution Optimisation Problems," Sixth International Conference on Hydroinformatics, Singapore.

Gibbs, M. S., Morgan, N., Maier, H. R., Dandy, G. C., Nixon, J. B., and Holmes, M. (2006). "Investigation into the relationship between chlorine decay and water distribution parameters using data driven methods." Mathematical and Computer Modelling, 44(5), 485-498.

GWI. (2007). "Chlorine's Glory Days." Global Water Intelligence.

Haestad Methods (2002) "*Automated Skeletonization Techniques*." Haestad Methods Inc., Waterbury, Connecticut, USA.

Halhal, D., Walters, G. A., Ouazar, D., and Savic, D. A. (1997). "Water network rehabilitation with structured messy genetic algorithm." *Journal of Water Resources Planning and Management*, 123(3), 137-146.

Halton, J. H. (1960). "On the Efficiency of Certain Quasi-Random Sequences of Points in Evaluating Multi-Dimensional Integrals." *Numerical Mathematics*, 2, 84-90.

Hammersley, J. M. (1960). "Monte Carlo Methods for Solving Multivariable Problems." *Annals of the New York Academy of Science*, 86, 844-874.

Hashimoto, T., Stedinger, J. R., and Loucks, D. P. (1982). "Reliability, Resiliency and Vulnerability Criteria For Water Resource System Performance Evaluation." *Water Resources Research*, 18(1), 14-20.

Hewitson, C., and Dandy, G. C. (2000). "Optimisation of Water Distribution Systems Including Hydraulic and Water Quality Criteria." *Proceedings, Water Network Modelling for Optimal Design and Management, International Symposium CWS2000,*, Exeter, UK, 195-204.

Jin, Y. C., Olhofer, M., and Sendhoff, B. (2002). "A framework for evolutionary optimization with approximate fitness functions." *Ieee Transactions on Evolutionary Computation*, 6(5), 481-494.

Jin, Y. (2005). "A comprehensive survey of fitness approximation in evolutionary computation." *Soft Computing*, 9(1), 3-12.

Jin, Y., and Branke, H. (2005). "Evolutionary Optimization in Uncertain Environments - A survey." *IEEE Transactions on Evolutionary Computation*, 9(3), 303-317.

Johnson, V. M., and Rogers, L. L. (2000). "Accuracy of neural network approximators in simulation-optimization." *Journal of Water Resources Planning and Management-Asce*, 126(2), 48-56.

Kalagnanam, J. R., and Diwekar, U. M. (1997). "An Efficient Sampling Technique for Off-line Quality Control." *Technometrics*, 39(3), 308-319.

Kapelan, Z. S., Savic, D. A., and Walters, G. A. (2005). "Multiobjective design of water distribution systems under uncertainty." *Water Resources Research*, 41(11).

Khomsi, D., Walters, G. A., Thorley, A. R. D., and Ouazar, D. (1996). "Reliability Tester for Water-Distribution Networks." *Journal of Computing in Civil Engineering-ASCE*, 10(1), 10-19.

Khu, S. T., and Werner, M. G. F. (2003). "Reduction of Monte-Carlo simulation runs for uncertainty estimation in hydrological modelling." *Hydrology and Earth System Sciences*, 7(5), 680-692.

Khu, S. T., Savic, D., Liu, Y., and Madsen, H. (2004). "A fast evolutionary-based metamodelling approach for the calibration of a rainfall-runoff model." *Trans. 2nd Biennial Meeting of the International Environmental Modelling and Software Society, iEMSs: Manno, Switzerland*.

Kleijnen, J. P. C., and Sargent, R. G., (2000), "A methodology for fitting and validating metamodels in simulation," *European Journal of Operational Research*, vol. 120, pp. 14-29.

Kourakos, G., and Mantoglou, A. (2009). "Pumping optimization of coastal aquifers based on evolutionary algorithms and surrogate modular neural network models." *Advances in water resources*, 32(4), 507-521.

Lansey, K. E., and Mays, L. W. (1989). "Optimization Model for Water Distribution System Design." *Journal of Hydraulic Engineering-ASCE*, 115(10), 1401-1418.

Leshno, M., Lin, V. Y., Pinkus, A., and Schocken, S. (1993). "Multilayer Feedforward Networks With a Nonpolynomial Activation Function Can Approximate Any Function." *Neural Networks*, 6, 861-867.

Lingireddy, S., and Ormsbee, L. E. (1998). "Neural Networks in Optimal Calibration of Water Distribution Systems." Artificial Neural Networks for Civil Engineers: Advanced Features and Applications, I. Flood and N. Kartam, eds., ASCE, 53-76.

Mackle, G., Savic, D. A., and Walters, G. A. (1995). "Application of genetic algorithms to pump scheduling for water supply." *Proceedings of the 1st IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications GALESIA '95, IEE Conference Publication, Sheffield, England. IEE, Stevenage, Engl, 400–405.*

Maier, H. R., Kapelan, Z., Kasprzyk, J., Matott, L. S., de Conceicao Cunha, M., Dandy, G., Gibbs, M. S., Keedwell, M. S., Marchi, A., Ostfeld, A., Savic, D., Solomatine, D., Vrugt, J. A., Zecchin, A. C., Minsker, B., Barbour, E., Kang, D., Kuczera, G., and Pasha, F. (2014). "Evolutionary Algorithms and Other Metaheuristics in Water Resources: Current Status, Research Challenges and Future Directions." *Environmental Modelling & Software*, submitted.

Maier, H. R., Simpson, A. R., Zecchin, A. C., Foong, W. K., Phang, K. Y., Seah, H. Y., and Tan, C. L. (2003). "Ant colony optimization distribution for design of water systems." *Journal of Water Resources Planning and Management-Asce*, 129(3), 200-209.

Martinez, F., Hernandez, V., Alonso, J. M., Rao, Z. F., and Alvisi, S. (2007). "Optimizing the operation of the Valencia water-distribution network." *Journal of Hydroinformatics*, 9(1), 65-78.

May, R. J., Dandy, G. C., Maier, H. R., and Nixon, J. B. (2008). "Application of partial mutual information variable selection to ANN forecasting of water quality in water distribution systems." Environmental Modelling & Software, 23(10), 1289-1299.

McKay, M. D., Beckman, R. J., and Conover, W. J. (1979). "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code." *Technometrics*, 21(2), 239-245.

Meckesheimer, M., Booker, A. J., Barton, R. R., and Simpson, T. W. (2002). "Computationally Inexpensive Metamodel Assessment Strategies." *AIAA Journal*, 40(10), 2053-2060.

Mugunthan, P., Shoemaker, C. A., and Regis, R. G. (2005). "Comparison of function approximation, heuristic, and derivative-based methods for automatic calibration of computationally expensive groundwater bioremediation models." *Water Resources Research*, 41(11).

Murphy, J. M., Sexton, D. M. H., Jenkins, G. J., Booth, B. B. B., Brown, C. C., Clark, R. T., Collins, M., Harris, G. R., Kendon, E. J., and Betts, R. A. (2009). "UK climate projections science report: climate change projections." Met Office Hadley Centre, Exeter, UK.

Murphy, L. J. (2014). "pers. comm." Senior Engineer, Optimatics.

Neelakantan, T. R., and Pundarikanthan, N. V., (2000). "Neural Network-Based Simulation-Optimization Model for Reservoir Operation," *Journal of Water Resources Planning and Management - ASCE*, vol. 126, pp. 57-64.

Optimatics (2014) "Optimizing Water Systems." http://www.optimatics.com, Accessed: March 28, 2014.

Ormsbee, L., and Kessler, A. (1990). "Optimal Upgrading of Hydraulic-Network Reliability." *Journal of Water Resources Planning and Management - ASCE*, 116(6), 784-802.

Ostfeld, A., and Salomons, S. (2005). "A hybrid genetic-instance based learning algorithm for CE-QUAL-W2 calibration." *Journal of Hydrology*, 310(1), 122-142.

Phelps, R. (2008). "Field Measurement of Total Residual Chlorine". SESDPROC-112-R1, US-Environmental Protection Agency, Science and Ecosystem Support Division.

Rao, Z. F., and Salomons, E. (2007). "Development of a real-time, near-optimal control process for water-distribution networks." *Journal of Hydroinformatics*, 9(1), 25-37.

Rastrigin, L. A. (1974). "Systems of extremal control." Nauka, Moscow.

Razavi, S., Tolson, B. A., and Burn, D. H. (2012a). "Review of surrogate modeling in water resources." *Water Resources Research*, 48(7).

Razavi, S., Tolson, B. A., and Burn, D. H. (2012b). "Numerical assessment of metamodelling strategies in computationally intensive optimization." *Environmental Modelling and Software*, 34(June), 67-86.

Rogers, L. L., and Dowla, F. U., (1994) "Optimization of Groundwater Remediation using Artificial Neural Networks with Parallel Solute Transport Modeling," *Water Resources Research*, vol. 30, pp. 457-481.

Sakarya, A. B. A. , and Mays, L. W., (2000) "Optimal Operation of Water Distribution Pumps Considering Water Quality," *Journal of Water Resources Planning and Management - ASCE*, vol. 126, pp. 210-220.

Salomons, E., Goryashko, A., Shamir, U., Rao, Z. F., and Alvisi, S. (2007). "Optimizing the operation of the Haifa-A water-distribution network." *Journal of Hydroinformatics*, 9(1), 51-64.

Sasena, M., Papalambros, P., & Goovaerts, P. (2002). "Global optimization of problems with disconnected feasible regions via surrogate modeling." In *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization* (4-6).

Savic, D. A., and Walters, G. A. (1997). "Genetic Algorithms for Least-Cost Design of Water Distribution Networks." *Journal of Water Resources Planning and Management-ASCE*, 123(2), 67-77.

Schaake, J. C. and Lai, F. H. (1969). "Linear Programming and Dynamic Programming Application to Water Distribution Network Design." *Report 116*, Department of Civil Engineering, Massachusetts Institute of Technology.

Shinstine, D. S., Ahmed, I., and Lansey, K. E. (2002). "Reliability/Availability Analysis of Municipal Water Distribution Networks: Case Studies." *Journal of Water Resources Planning and Management-ASCE*, 128(2), 140-151.

Shoemaker, L., Lai, F.-H., Zhen, J. X., Alvi, K., Riverson, J., and Rafi, T. (2008). "Optimizing BMP Placement at Watershed-Scale Using SUSTAIN." EWRI 2008 World Environmental and Water Resources Congress, Honolulu, HI, USA.

Simpson, A. R., Dandy, G. C., and Murphy, L. J. (1994). "Genetic Algorithms Compared to Other Techniques for Pipe Optimization." *Journal of Water Resources Planning and Management - ASCE*, 120(4), 423-443.

Simpson, T. W., Lin, D. K. J., and Chen, W. (2001). "Sampling strategies for computer experiments: design and analysis." *International Journal of Reliability and Applications*, 2(3), 209-240.

Su, Y., Mays, L. W., Duan, N., and Lansey, K. E. (1987). "Reliability-Based Optimization Model for Water Distribution Systems." *Journal of Hydraulic Engineering - ASCE*, 113(12), 1539-1556.

Surjanovic, S., and Bingham, D. (2013). "Virtual Library of Simulation Experiments: Test Functions and Dataset." http://www.sfu.ca/~ssurjano, Accessed: March 23, 2014.

Tolson, B. A., Maier, H. R., and Simpson, A. R. (2001). "Water Distribution Network Reliability Estimation Using the First-Order Reliability Method." *Proceedings of the World Water and Environmental Resources Congress*, Orlando, Florida, On CD ROM.

Tolson, B. A., Maier, H. R., Simpson, A. R., and Lence, B. J. (2004). "Genetic algorithms for reliability-based optimization of water distribution systems." *Journal of Water Resources Planning and Management-Asce*, 130(1), 63-72.

Turner, C. J., Campbell, M. I., & Crawford, R. H. (2003). "Generic sequential sampling for metamodel approximations." In *ASME 2003 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (555-564). American Society of Mechanical Engineers.

Tyagi, A. (2003). "A Method Ensuring Residual Chlorine in Water Distribution System." *World Water & Environmental Resources Congress*, Philadelphia, PA, USA, On CD-ROM.

Ulanicki, B., Zehnpfund, A. and Martinez, F. (1996) "Simplification of water network models", *Proceedings of the Hydroinformatics 96 International Conference, International Association for Hydraulic Research*, ETH Zurich, September 9-13.

Van Zyl, J. E., Savic, D. A., and Walters, G. A. (2004). "Operational optimization of water distribution systems using a hybrid genetic algorithm." *Journal of Water Resources Planning and Management*, 130(2), 160–170.

Walski, T. (2001). "The wrong paradigm - why water distribution optimization doesn't work." *Journal of Water Resources Planning and Management-ASCE*, 127(4), 203-205.

Walski, T. M., Downey Brill, J. E., Gessler, J., Goulter, I. C., Jeppson, R. M., Lansey, K., Lee, H.-L., Liebman, J. C., Mays, L., Morgan, D. R., and Ormsbee, L. (1987). "Battle of the Network Models: Epilogue," *Journal of Water Resources Planning and Management - ASCE*, vol. 113, pp. 191-203, 1987.

Walters, G. A., Halhal, D., Savic, D. A., and Ouazar, D. (1999). "Improved design of "Anytown" distribution network using structured messy genetic algorithms." *Urban Water*, 1(1), 23-38.

Wu, W., Dandy, G. C., and Maier, H. R. (2014). "Protocol for developing ANN models and its application to the assessment of the quality of the ANN model development process in drinking water quality modeling." Environmental Modelling and Software, 54, 108-127.

Wu, Z. Y., and Simpson, A. R., (2001). "Competent Genetic-Evolutionary Optimization of Water Distribution Systems," *Journal of Computing in Civil Engineering-ASCE*, vol. 15, pp. 89-101.

Xu, C., and Goulter, I. G. (1999). "Reliability-Based Optimal Design of Water Distribution Networks." *Journal of Water Resources Planning and Management - ASCE*, 125(6), 352-362.

Yan, S., and Minsker, B. (2006). "Optimal groundwater remediation design using an Adaptive Neural Network Genetic Algorithm." *Water Resources Research*, 42(5), 05407.

Yan, S., and Minsker, B. (2011). "Applying dynamic surrogate models in noisy genetic algorithms to optimize groundwater remediation designs." *Journal of Water Resources Planning and Management*, 137(3), 284-292.

Zecchin, A., Simpson, A. R., Maier, H. R., and Nixon, J. B., (2005) "Parametric Study for an Ant Algorithm Applied to Water Distribution System Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 9, pp. 175-191, 2005.

Zhang, X., Srinivasan, R., and Van Liew, M. (2009). "Approximating SWAT Model Using Artificial Neural Network and Support Vector Machine1." *JAWRA Journal of the American Water Resources Association*, 45(2), 460-474.

Appendix A:

Systematic Approach applied to

Simple Mathematical Functions

In this section, the proposed process is illustrated by way of two mathematical functions. It should be noted that these two functions would not need to be replaced by metamodels in practice and are used purely for the sake of demonstration.

Firstly, consider Bukin's function N6 (Surjanovic and Bingham 2013), as given by Eq. A.1.

$$\min z = f(\boldsymbol{x}) = 100\sqrt{|x_2 - 0.01x_1^2|} + 0.01|x_1 + 10| \tag{A.1}$$

That may be broken down into a series of calculation steps that can be assessed in terms of computational requirements, dimensionality and smoothness, as given in Table A.1.

Table A.1. Assessment of calculation steps of Bukin's function N6.

	Calculation	Computational Assessment	Dimensionality Assessment	Smoothness Assessment
1	$y_1 = x_1 + 10$	Trivial	Low	Smooth
2	$y_2 = 0.01\|y_1\|$	Trivial	Low	Non-smooth
3	$y_3 = 0.01x_1^2$	Trivial	Low	Smooth
4	$y_4 = x_2 - y_3$	Trivial	Low	Smooth
5	$y_5 = \|y_4\|$	Trivial	Low	Non-smooth
6	$y_6 = 100\sqrt{y_5}$	Trivial	Low	Smooth
7	$z = y_2 + y_6$	Trivial	Low	Smooth

Because this is a simple mathematical function, the computational assessment is trivial for each step and the dimensionality is low, hence S_1 and S_2 should each include all calculation steps. These are given in Eqs A.2 and A.3, respectively.

There are two non-smooth calculation steps, as they include absolute value functions, which are more difficult for metamodels to approximate than smoother functions, hence they should not be included in the metamodel scope. Therefore there are three potential metamodel scopes for S3 (recall that a metamodel scope must include connected calculation steps only). These are given by Eq. A.4.

Determining the best overall metamodel scope is then a matter of applying Eq. 5.6, which results in Eq. A.5 for this example, i.e. there are three possible metamodel scopes. The metamodels' input and output variables (for the three possible metamodel scopes) are then defined in Eqs. A.6 and A.7, respectively. The metamodel should act as a surrogate for either calculation step 1, or steps 3 and 4, or steps 6 and 7.

$$S_1 = \{f_i\}, i = 1, \dots, 7 \tag{A.2}$$

$$S_2 = \{f_i\}, i = 1, \dots, 7 \tag{A.3}$$

$$S_3 = \{f_1\}\ OR\ \{f_3, f_4\}\ OR\ \{f_6, f_7\} \tag{A.4}$$

$$S_{Best} = \bigcap_{i=1}^{3} S_i = \{f_1\}\ OR\ \{f_3, f_4\}\ OR\ \{f_6, f_7\} \tag{A.5}$$

$$\boldsymbol{x_{MM}} = \{x_1\}\ OR\ \{x_1, x_2\}\ OR\ \{y_2, y_5\} \tag{A.6}$$

$$\boldsymbol{y_{MM}} = \{y_1\}\ OR\ \{y_4\}\ OR\ \{z\} \tag{A.7}$$

Secondly, consider the 20-dimensional version of Rastrigin's function (Rastrigin 1974), given by Eq. A.8, which may be broken into five calculation steps, which are assessed in Table A.2.

$$\min z = f(\boldsymbol{x}) = \sum_{i=1}^{20} \left(x_i^2 - cos(2\pi x_i)\right) \tag{A.8}$$

Table A.2. Assessment of calculation steps of Rastrigin's function.

	Calculation	Computational Assessment	Dimensionality Assessment[a]	Smoothness Assessment
1	$y_i = 2\pi x_i, i = 1, \dots, 20$	Trivial	Trivial for each y_i	Smooth
2	$y_{i+20} = x_i^2, i = 1, \dots, 20$	Trivial	Trivial for each y_{i+d}	Smooth
3	$y_{i+40} = cos(y_i),$ $i = 1, \dots, 20$	Trivial	Trivial for each y_{i+2d}	Smooth
4	$y_{i+60} = y_{i+d} y_{i+2d},$ $i = 1, \dots, 20$	Trivial	Trivial for each y_{i+3d}	Smooth
5	$z = \sum_{i=1}^{20} y_{i+60}$	Low	Medium	Smooth

The possible metamodel scopes according to each criterion are presented in Eqs. A.9-A.11. Upon examination, the intersection of the three scopes is the null set, {0}, hence the best metamodel must be determined by considering two criteria, as given by Eqs. A.12- A.13. Note $S_{1,2} = \{0\}$.

$$S_1 = \{f_5\} \tag{A.9}$$

$$S_2 = \{f_1, f_2, f_3, f_4\} \tag{A.10}$$

$$S_3 = \{f_1, f_2, f_3, f_4, f_5\} \tag{A.11}$$

$$S_{1,3} = \{f_5\} \tag{A.12}$$

$$S_{2,3} = \{f_1, f_2, f_3, f_4\} \tag{A.13}$$

Consequently, there are two best metamodel scopes according to this process, $S_{1,3}$ *and* $S_{2,3}$. Therefore both will need to be considered in the overall metamodelling process; at least up until calibration (step 6 of Figure 5-1). The input and output variables for the two metamodels are given in Eqs. A.14 and A.15. The metamodel should act as a surrogate for either calculation step 5, or steps 1-4.

$$\boldsymbol{x_{MM}} = \{y_{i+3d}\}\ OR\ \{x_i\}, i = 1, \dots, d \tag{A.14}$$

$$\boldsymbol{y_{MM}} = \{z\}\ OR\ \{y_{i+3d}\}, i = 1, \dots, d \tag{A.15}$$

Appendix B:

Hammersley Sampling for

Stochastic Variables

Hammersley samples (Hammersley 1960), as modified by Halton (1960), are generated according to Eq. B.1.

$$s(i,n) = \sum_{j=0}^{k-1} \left[\left\lfloor ip^{-j} \right\rfloor - \left\lfloor \frac{\left\lfloor ip^{-j} \right\rfloor}{p} \right\rfloor p \right] (p^{-j-1}) \tag{B.1}$$

Where *s(i,n)* is the *i*-th sample in the *n*-th dimension; *p* is the *n*-th prime number (starting at 2); *k* is given by equation B.2; and ⌊ ⌋ is the floor function (i.e. round down to the nearest integer).

$$k = \left\lceil \frac{ln(N_{MCS})}{ln(p)} \right\rceil \tag{B.2}$$

Where N_{MCS} is the total number of samples that will be generated; and ⌈ ⌉ is the ceiling function (i.e. round up to the nearest integer). A key advantage of HS over Latin Hypercube Sampling (LHS) (McKay et al. 1979), which is a more commonly used sampling approach, is that N_{MCS} can be increased for subsequent samples without affecting previous samples in cases where additional samples need to be generated.

HS was designed to be a "low-discrepancy" sampling method (Hammersley 1960), where discrepancy can be considered as a measure of the largest unsampled area across *n* dimensions. This is demonstrated in the following figures. Figure B.1 shows examples of data generated from a uniform distribution (a) randomly, (b) using LHS with 4 stratifications, (c) using HS, comparing dimensions 1 and 2, and (d) using HS, comparing dimensions 5 and 10. Figure B.2 shows the same samples, but transformed into a standard normal distribution. It can be seen that there is a regular pattern to HS and that the points are close to evenly distributed (for the uniform case). This is a feature of HS, rather than the outcome of a fortunate random seed, as HS is deterministic, rather than stochastic. For further details of HS in

metamodelling applications and a comparison to other data generation methods, the reader is referred to Kalagnanam and Diwekar (1997) and Simpson et al. (2001).

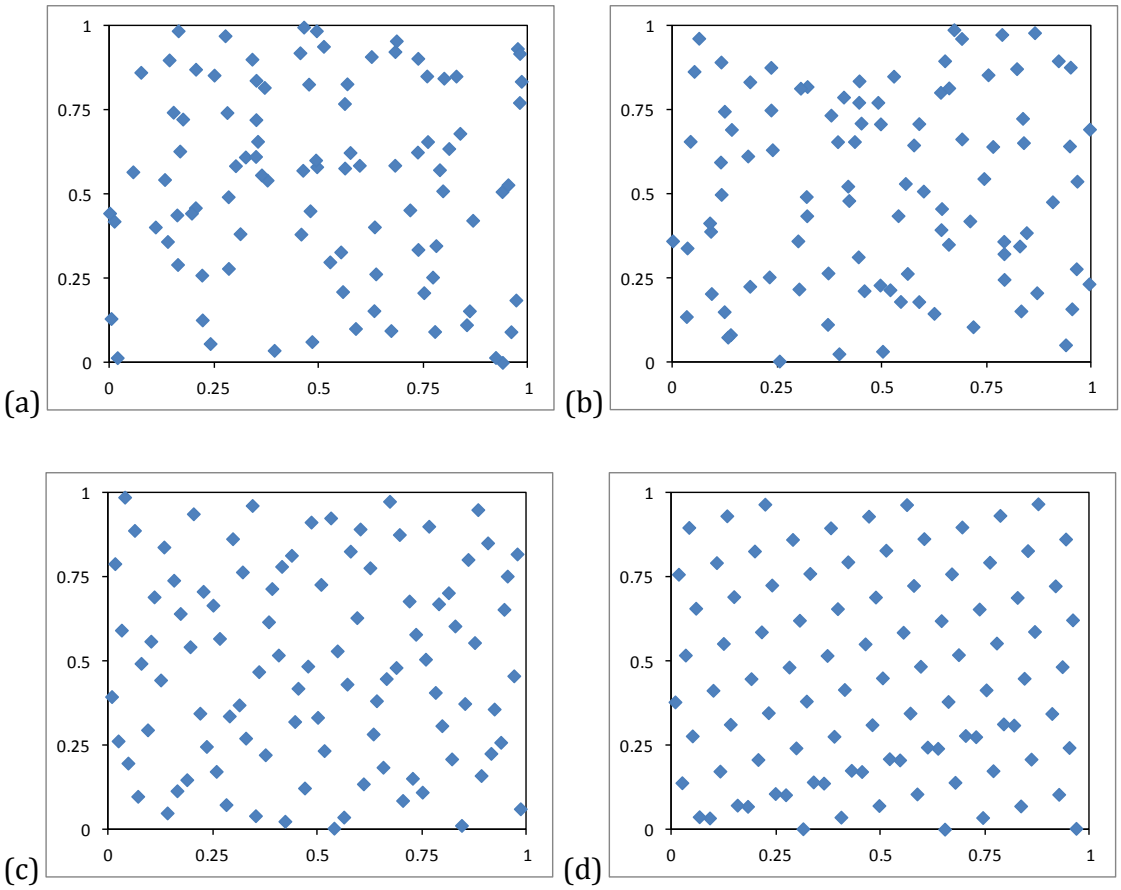


Figure B.1. Example of data sampling methods in 2 dimensions (96 points in U[0,1]). (a) RS, (b) LHS with 4 stratifications per dimension, (c) HS (dimension 1 and 2), (d) HS (dimension 5 and 10).

One negative aspect of HS is that for high dimensions (> ~20), the number of samples that must be obtained to ensure even coverage of the sample space becomes quite large. That is not a problem for WDS optimization where there are few key sources of uncertainty (i.e. demand, pipe roughness and chlorine decay rate) and these can reasonably be considered to be highly spatially correlated; however, the reader should be wary when considering HS for other applications.

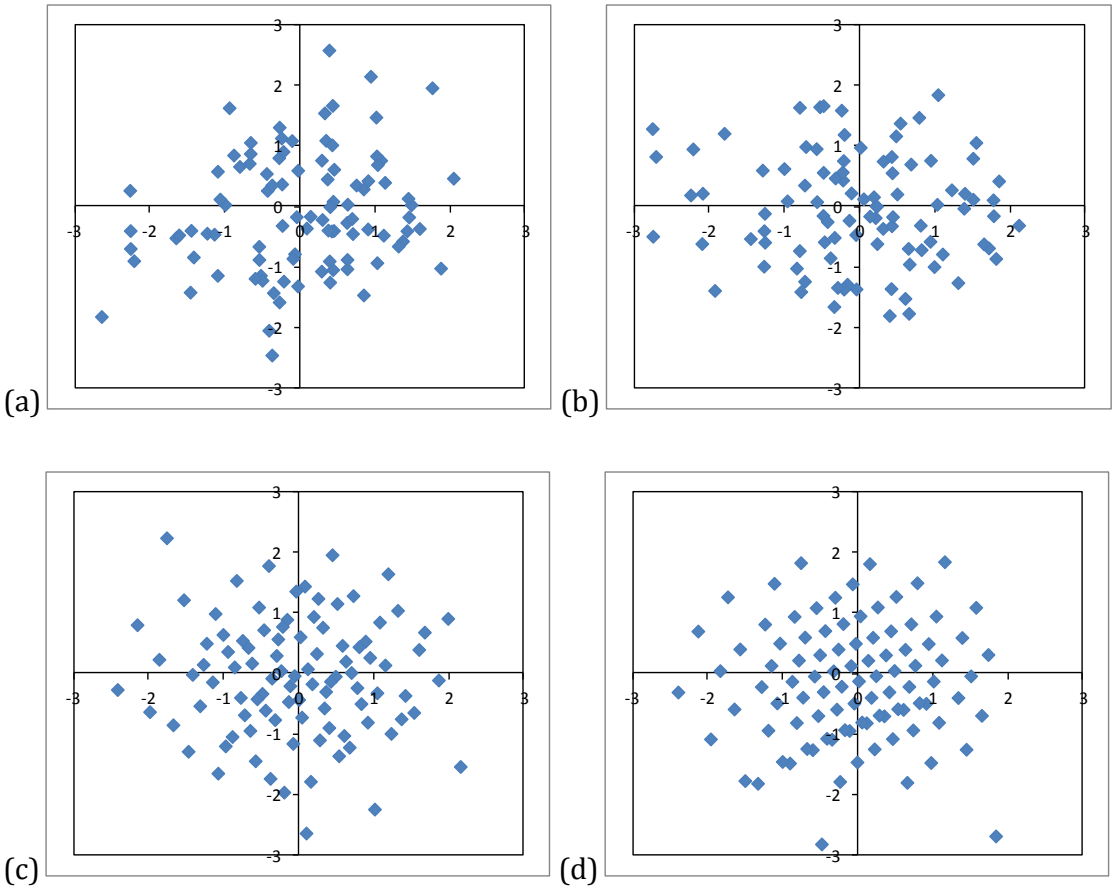


Figure B.2. Example of data sampling methods in 2 dimensions (96 points in N[0,1]). (a) RS, (b) LHS with 4 stratifications per dimension, (c) HS (dimension 1 and 2), (d) HS (dimension 5 and 10).

Appendix C:

Case Study Details

C.1 New York Tunnels

For details on NYT, the reader is referred to Maier et al. (2003). The reader is also referred to Broad et al. (2005) for details pertaining to the modification of NYT for the purposes of water quality modelling (including chlorine dosing decision variables, minimum chlorine residual constraints and the addition of a demand pattern for use in an EPS).

Further modifications to NYT have been made for this paper. Disinfection costs have been included by quantifying the chlorine dosed and a net present value analysis is used. It was assumed the design flow in the original NYT model represents the peak hour demand, however, NPV chlorine costs should be based on the average day demand. Therefore peaking factors were assumed for this conversion, as shown in Table C.1. Table C.1 also shows the assumed NPV parameters. Chlorine costs from 2007 (55c/kg, (GWI 2007)) were converted to 1969 prices (8.6c/kg) for reasonable comparison with pipe costs assuming a CPI rate of 5% p.a.

Table C.1. Assumed data for calculating disinfection costs for NYT.

Parameter	Value
Peak hour factor	1.3
Peak day factor	3
Chlorine cost	8.6 c/kg
Discount rate	6%
Design life	20

A summary of the simulation model statistics is provided in Table C.2 and a summary of the definition of the search space is provided in Table C.3.

Table C.2. Simulation model summary for the New York Tunnels case study.

Model Component	Number
Pipes	42
Sources	1
Junctions	19
Chlorine Dosing	1

Table C.3. Summary of optimization decisions and search space for the New York Tunnels case study.

Property	Number
Pipe Decisions	21
Chlorine Dosing	1
Options per Pipe	16
Options per Dose	21
Search Space	16^{21}x21 = 4.8x10^{27}

The NYT problem was also modified to include randomness in demand, pipe roughness and chlorine decay rate. Mean values were calculated using Eqs. C.1 and C.2, based on the assumption that original values would have been selected conservatively, and as such the values from the original problem were not assumed as the mean values.

$$DM_{\mu} = \widehat{DM} - 2DM_{\sigma} \quad \text{(C.1)}$$

$$C_{\mu} = \hat{C} + 2C_{\sigma} \quad \text{(C.2)}$$

Where DM_{μ} and DM_{σ} are the mean and standard deviation nodal base demand values used in this case study, respectively; $\widehat{DM}$ is the nodal base demand from the original problem; C_{μ} and C_{σ} are the mean and standard deviation Hazen-Williams

roughness coefficients used in this case study, respectively; and $\hat{C}$ is the Hazen-Williams roughness coefficient from the original problem.

Each property was assumed to have a normal distribution with a coefficient of variation of 0.1. Therefore, the mean demand and Hazen-Williams roughness coefficients can be calculated using Eqs. C.3-C.4.

$DM_{\mu} = \frac{\widehat{DM}}{1.2}$	(C.3)
$C_{\mu} = \frac{\hat{C}}{0.8}$	(C.4)

The risk metrics shown in Table C.4 were used as constraints. The EPANet input file for this problem can be downloaded as supplementary material (Broad 2014a).

Table C.4. Risk metric constraints used for NYT case study.

Risk-Metric Constraint	Value
Hydraulic Reliability	95%
Hydraulic Vulnerability	0.5 m
Water Quality Reliability	95%
Water Quality Vulnerability	0.1 mg/L

C.2 Pacific City

A schematic of Pacific City is presented in Figure C.1. All reservoirs and pipes are identifiable. Additionally, pipe decisions are shown as thick black lines, while chlorine dosing locations (primarily at reservoirs) are shown as crosses. All pipe decisions are duplication options, which had been previously identified. A summary of the model statistics is provided in Table C.5 and the EPANet input file for this problem can be downloaded as supplementary material (Broad 2014b).

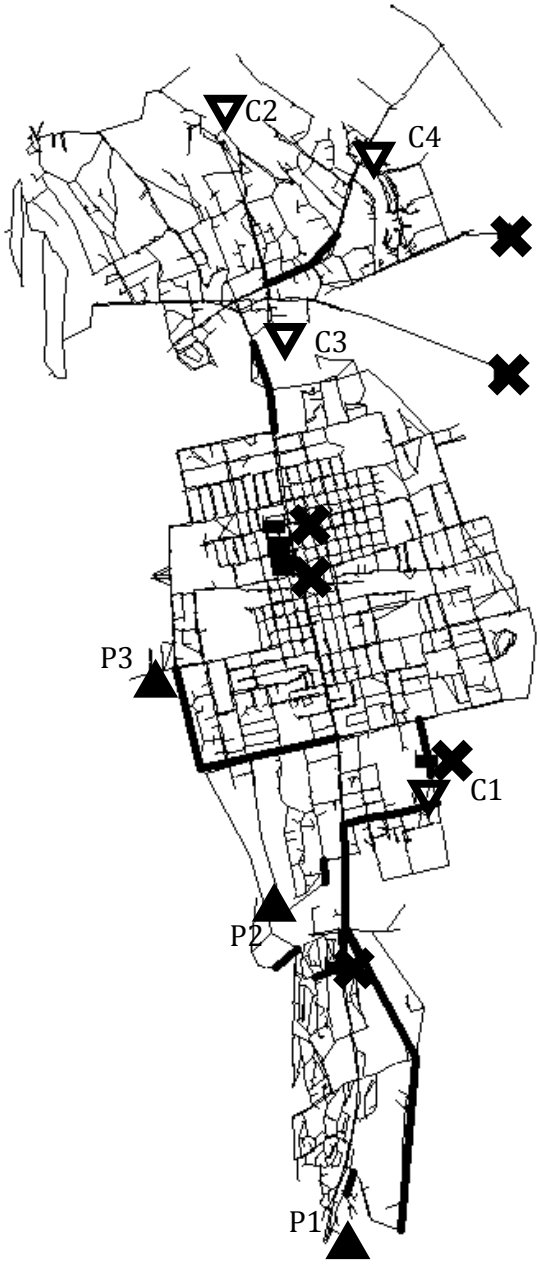


Figure C.1. Schematic of the Pacific City case study.

Table C.5. Simulation model summary of the Pacific City case study.

Model Component	Number
Pipes	6944
Sources (groundwater)	9
Junctions	8715
Valves (isolation)	2716
Unique demand patterns	18
Chlorine Dosing	6

The optimization formulation is summarized in the following tables. Table C.6 shows the pipe decision options and their unit cost rates. The unit cost for chlorine dosing is 20 c/kg and each dosing location has 21 options, ranging from 0.5 to 2.5 mg/L, increasing in increments of 0.1 mg/L. It was observed that there was very little variation in flows at the sources for the different solutions, hence representative flows were used at each dosing location in order to calculate the chlorine mass dosed. A summary of the decisions is given in Table C.7. Minimum pressures were 25 m of head and minimum chlorine residuals were 0.5 mg/L. The same risk metric thresholds were used as for NYT (see Table C.4).

Table C.6. Pipe decision options for the Pacific City case study.

Diameter [mm]	Cost [$/m]
150	401.8
175	429.2
200	444.2
225	473.9
250	490.8
275	513.7
300	538.3
325	571.0
350	588.1
375	612.7
400	649.6
425	673.4
450	696.5
475	731.0
500	760.5
525	781.0
550	820.2
575	847.2

Table C.7. Summary of optimization decisions and search space for the Pacific City case study.

Property	Number
Pipe Decisions	23
Chlorine Dosing	6
Options per Pipe	18
Options per Dose	21
Search Space	$18^{23}x21^{6} = 6.4x10^{36}$