Computational Intelligence in Rainfall–Runoff Modeling

Computational Intelligence in Rainfall–Runoff Modeling

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. dr. ir. J. T. Fokkema,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen

op maandag 16 november 2009 om 12:30 uur

door

Nelis Jacob DE VOS

civiel ingenieur
geboren te Gorinchem.

Dit proefschrift is goedgekeurd door de promotor:

Prof. dr. ir. H. H. G. Savenije

Samenstelling promotiecommissie:

Rector Magnificus	voorzitter
Prof. dr. ir. H. H. G. Savenije	Technische Universiteit Delft, promotor
Prof. dr. H. V. Gupta	University of Arizona
Prof. dr. E. Zehe	Technische Universität München
Dr. ing. T. H. M. Rientjes	ITC Enschede
Prof. dr. D. P. Solomatine	UNESCO–IHE, Technische Universiteit Delft
Prof. drs. ir. J. K. Vrijling	Technische Universiteit Delft
Prof. dr. ir. A. W. Heemink	Technische Universiteit Delft

Reservelid:

Prof. dr. ir. G. S. Stelling	Technische Universiteit Delft

ISBN 978-90-8559-585-4

Keywords: hydrological modeling, computational intelligence, artificial neural networks, model calibration, evolutionary algorithms, clustering.

Printed by Optima Grafische Communicatie

"The important thing in science is not so much to obtain new facts as to discover new ways of thinking about them."

Sir William Lawrence Bragg

Preface

The workings of nature have always puzzled mankind, and its explanations of the physical world range from questionable but ingenious to acceptable and elegant. After merely some years of following the noble tradition of seeking such explanations, most likely of the former kind, I can nevertheless say that the view from the shoulders of giants is majestic. The late Carl Sagan wrote: science is a way of thinking much more than it is a body of knowledge. Indeed, I feel that my perspective on things has changed. Think of putting on a pair of glasses that let you see this world somewhat clearer and easier on the eyes. Their design is quite peculiar, though, and you can never take them off.

One of the things that I have learned during my Ph. D. research is that prediction is a tricky thing, fraught with errors and uncertainty. Should I have known this in 2003, I might have been more critical of my predictions that I would not regret pursuing a doctorate degree and that it would be actually possible to get one. Fortunately, I was right about both, which is partly thanks to the excellent conditions set by Delft University of Technology, which funded the first two years of this research through the DIOC programme "Transient processes in Hydraulic Engineering and Geohydrology", and its Water Resources Section, which financed the remaining two-and-a-half years.

The Royal Meteorological Institute of Belgium (Brussels, Belgium) and the National Weather Service Hydrology Laboratory (MD, USA) are gratefully acknowledged for providing the hydrometeorological data that were necessary for my work. I was also helped by having the following software codes kindly made available to me by their authors: the Genetic Algorithm Optimization Toolbox by Chris Houck, Jeff Joines and Mike Kay [1], the MOSCEM–UA algorithm by Jasper Vrugt, the Differential Evolution algorithm by Rainer Storn [2], its self-adaptive variant by Janez Brest, and the HyMod model by Hoshin Gupta.

Of course the completion of a Ph. D. thesis requires much more than financial resources, data and software. For several essential — but at times elusive — phenomena such as insight, inspiration and motivation I have relied on many people, some of whom I would like to mention here.

1. http://www.ise.ncsu.edu/mirage/GAToolBox/gaot/
2. http://www.icsi.berkeley.edu/~storn/code.html

The most valuable contributor to this work is Tom Rientjes. It has been many years since I first stepped into your office, Tom. A lot of things have changed since then, but never our mutual appreciation and respect. Those long and exhausting discussions during which we together pushed this work to its present level will be sorely missed! I owe much to you for my professional development, and I offer you my deepest gratitude for this and for our friendship.

My promotor, Huub Savenije, has always supported me unconditionally. Huub, I sincerely thank you for the freedom you provided me with, for your guidance and good advice, and for the way you were able to make me regain my interest in, shall I say, *pure* hydrology.

In 2007, I received a travel grant from the Netherlands Organization for Scientific Research (NWO) to stay for three months at the University of Arizona to discuss my work with Hoshin Gupta. It has been a great privilege to meet you, Hoshin, and to have benefited from your hospitality and guidance. Your philosophy and overview were crucial in putting the pieces of my puzzle together.

I am grateful to Dimitri Solomatine for allowing me to follow his inspiring course on hydroinformatics at UNESCO–IHE in 2004.

My wonderful colleagues in Delft have made that all-too-familiar environment a pleasant and fruitful one for doing my research. My paranymphs Miriam Gerrits and Steven Weijs, Fabrizio Fenicia, Zhang Guoping, Nguyen Ahn Duc, Robert Kamp, and all other Ph. D. students and staff members I thank for entertaining discussions, helpful tips and lots of fun. I nevertheless strongly benefited from having a change of scenery by staying in the U.S.A. I wish to thank all my friends overseas, and Koray Yilmaz in particular, for all the things I experienced and learned in Tucson and San Francisco.

My parents have enabled and encouraged my education, perhaps more than I often did. Mom and dad, thank you for your love and support.

The final phase of my research was not always easy due to the overlap with a new job, but fortunately one of the last things I did in Delft was meeting the lovely Zhao Yi. Thanks for cheering me up and motivating me when I needed it, baby.

Time for a confession from this ‘hydrologist’: I have only once played the role of experimentalist during the entire period of my Ph. D. research. (To all the people that subsequently want to accuse me of being afraid to get my hands dirty, I say this: you are clearly blissfully unaware of the average number of bacteria that live on a computer keyboard.) My justification is that I was captivated by the beauty of applying computational intelligence to hydrology. Being confronted with the paradigms and techniques of two research fields and realizing their similar, different and complementary aspects have taught me a lot. Speaking more broadly, I have found that being a part of the scientific community also allows the upscaling of one’s thinking. It is comforting to join scientists

from all backgrounds, countries and cultures in discussing in the common language of science our collective efforts to better understand the world we live in, be it at conferences or in the literature. Both have been excellent places for retreats whenever I needed to take a step back and find a new perspective on my work. The iterative loop of questioning and revising one's own assumptions, methods or presentation can be endless — the question rises when to stop and accept the work as finished. Accepting deadlines for my research has always been slightly difficult for me, perhaps because in science there always seem to be significant improvements lurking around the corner. Not entirely unfortunately, such a deadline has presented itself for this thesis, so I beg the reader to judge mildly its contents.

Nico de Vos
September 2009, Schiedam

Contents

Chapter 1

Introduction

1.1 Background and Motivation

1.1.1 Rainfall–Runoff Modeling

Challenges Regarding Hydrological Models and Data

The transformation of precipitation over a river basin into river streamflow is the result of many interacting processes which manifest themselves at various scales of time and space. This is due to the variety and the heterogeneity of media in which water travels. The interaction of hydrology with other research disciplines such as atmospheric sciences, ecology and geology showcases just how diverse these media can be. The highly dynamic, nonlinear nature of most catchment systems is a reflection of the complex interaction between the various processes at different scales. Clearly, effective system descriptions are therefore not easily defined.

The issues of scale and heterogeneity make the definition of straightforward descriptions and models of hydrological processes very challenging, and also complicate the gathering of representative data from a catchment. Not only are measurements seldom without error, the difficult questions present themselves of how well observed data represent system behavior and how they translate to the quantities defined in the model representation of the system.

No laboratory experiment can be constructed in which the complexity of a natural hydrological system is adequately replicated, which is why researchers focus on simulation models of river basins. The challenge of Rainfall–Runoff (R–R) modeling originates from the combination of the complexity of a catchment system and the difficulty to properly and quantitatively express the information that is available about it. For these reasons, R–R modeling is considered one of the greatest challenges in hydrology, even after more than a century of research.

Why Model the Rainfall–Runoff Transformation?

The R–R transformation involves many processes that are at the very core of hydrology. R–R models that simulate these processes can therefore be used to advance the science, because the accordance or conflict with observed values of models based on existing hydrological theories can confirm or negate these theories.

Another, more practical reason to model the R–R transformation is for prediction purposes, ultimately to improve the quality or effectiveness of decisions related to water management issues. Some examples of areas in which streamflow information is needed are water resources assessment, flood protection, mitigation of droughts, channel or hydraulic construction design, assessment of contamination effects, ecological studies, and climate change impact assessments.

Historical Overview

The first models that predicted runoff from rainfall were developed as early as halfway the nineteenth century (e.g., the rational method by Mulvaney [1851]). Engineers interested in design criteria for constructions based these theories on empirically derived relationships and largely ignored the nonlinearities in R–R processes. Other methods were more focused on the routing of runoff such as the well-known unit hydrograph method. A large number of variations of this method exist that are all based on the idea of using a transfer function to calculate runoff from effective rainfall (e.g., the Nash-cascade model).

The problem of estimating how much of the rainfall effectively contributes to the runoff has always been the biggest challenge for modelers. As hydrological process understanding and the possibilities offered by digital computers grew, they turned to simulation of river basin behavior using so-called conceptual R–R models. Such models are based on a simple mass balance and simplified descriptions of hydrological processes and media. Despite the physical foundations of conceptual models, their parameters still need to be calibrated. A very large number of conceptual models exist, of which popular examples are the American Sacramento model, the Scandinavian HBV model, and the Japanese Tank model.

Physically-based and spatially distributed models of hydrological systems (e.g., SHE and IHDM models) became more popular in the 1970s and 1980s with the advancements in both computer power and accessibility. However, the large data demand and the complexity of calibration and simulation are important reasons for the lack of popularity of these models, even today. There are, however, simplified distributed models such as TOPMODEL that attempt to strike a balance between complexity and practicality.

In the last two decades, the R–R model approaches mentioned above have still been under development. With the development and application of modern data analysis, modeling and calibration techniques, the empirical approach has broadened to what can be called

data-driven R–R modeling. Conceptual models have also advanced because of better process and model understanding and improved calibration strategies. Finally, increasing computer power and new data sources such as remote sensing have helped to push the boundaries of physically-based models.

Current Developments

Having explored the various modeling possibilities in the past, the awareness that each model approach has its advantages and disadvantages has settled among hydrologists. Conceptual models often prove to be effective because they offer a combination of simplicity, transparency and good performance. Physically-based models are preferred when, for example, predicting the effect of changes in a river basin. Finally, for short-term streamflow forecasting, the accuracy of data-driven models is often unrivaled.

Nevertheless, the search for better R–R models is ongoing. The main drivers for developments in this field are:

1. Improved insights into both small-scale and large-scale processes of river systems (and the interactions between them),
2. Increase in quantity and quality of data through new and enhanced measurement techniques, along with improved abilities to extract information from data,
3. New views and paradigms in hydrological modeling methodology and philosophy,
4. Development of new modeling, calibration and data assimilation techniques.

Section 1.2 discusses the methodological and philosophical framework that has directed this research (cf. point 3). This framework has some overlap with point 2 as well, as one of the main ideas is to make better use of the information that is contained in the data. The main driving force of this work, however, comes from point 4 in the form of computational intelligence techniques.

1.1.2 Computational Intelligence

The field that is nowadays commonly called Computational Intelligence (CI) in fact evolved from various research fields such as soft computing [Zadeh, 1994], machine learning [Mitchell, 1997], evolutionary computing [Jong, 2006]. Nowadays, it also includes newer techniques from, for example, chaos theory and swarm intelligence. In recent years, CI has therefore increasingly become a generic term for a large diversity of techniques that use — often a weak sense of — intelligence in their approach. What prevents CI from being merely a hotchpotch of algorithms is the collective trait of its techniques to intelligently solve complex computational problems in science and technology [Palit and Popovic, 2005]. Solomatine [2005] discusses the similarities and differences between various research areas related to computational intelligence.

CI is an emerging field in which theoretical developments are still rapidly evolving and feedback from practical application is much needed. It is still unclear to what extent CI will change the field of simulation modeling but both the similar and complementary aspects of CI techniques and human problem-solving suggest a large promise. Examples of similarities include the way artificial neural networks mimic the functioning of the brain (albeit very rudimentary) and the use of fuzzy information such as words in a problem-solving framework [Zadeh, 1996]. An important complementary aspect is the increased computational ability of digital computers compared to human brains. The fact that computer power increases still strengthens the belief that CI can lead to significant developments in many fields of science.

In this work, two typical current exponents of CI are investigated: artificial neural networks and evolutionary algorithms. Both techniques have been applied in hydrological modeling in recent years and prove to be valuable alternatives to traditional approaches (more on this in Chapter 2). In the hydrological research community there exists a clear and urgent need to further investigate the application of such techniques in hydrological modeling in order to explore their potential value and pitfalls, and to formulate guidelines regarding their application.

1.2 Modeling Frameworks

In this section several frameworks from (hydrological) modeling are briefly presented that serve as a philosophical–methodological foundation for this work.

1.2.1 Systems Approach

A system is a theoretically defined set of components that interact or are in some way interdependent. The system is defined by the choices of system boundaries, and of which components and interactions to consider. A hydrological catchment is classified as an open system, meaning that the system interacts with an environment (e.g., a meteorological system) in the form of exchange of mass and energy. After the system is defined, a model of it can be built. According to Gupta *et al.* [2008], a model is

> *a simplified representation of a system, whose twofold purpose is to enable reasoning within an idealized framework and to enable testable predictions of what might happen under new circumstances*

where

> *the representation is based on explicit simplifying assumptions that allow acceptably accurate simulations of the real system.*

In the systems approach to R–R modeling, the modeler sees the hydrological catchment from a holistic instead of a reductionist viewpoint. The overall complexity of the system is considered a reason for not trying to separate its elements and processes but to regard the system as the only appropriate level of complexity on which to model or from which to deduce knowledge. Often this means that a so-called black-box approach is taken, in which the processes and elements of the systems are regarded unknown — although they not need to be. The only way of understanding the system is then to consider its input and output, and attempt to relate the two. From this point of view, building a model does not necessarily require knowledge of the physical principles involved.

1.2.2 Top-down Modeling

In the reductionist or bottom-up approach to modeling, a model is built starting from a detailed description of its basic elements. These elements are often specified on the basis of detailed experiments. A model can subsequently be built by linking the various (sub-systems of) elements together until the appropriate level of system detail. The bottom-up way of thinking has long been the dominant paradigm in hydrological modeling, but recent literature shows that the top-down paradigm is considered a promising alternative by many (e.g. Sivapalan *et al.* [2003b]; Young [2003]). The top-down paradigm follows the opposite direction of bottom-up modeling, and is defined by Klemeš [1983] as the route that

> *starts with trying to find a distinct conceptual node directly at the level of interest (or higher) and then looks for the steps that could have led to it from a lower level.*

This paradigm strongly relates to the systems approach, although the latter does not imply explaining the working of the system in terms of internal characteristics or processes at finer scales [Sivapalan *et al.*, 2003b].

1.2.3 Model Evaluation

Historically, there have been several changes in what is implied when hydrological modelers stated that they want to improve their models. Compare, for example, the issues raised on numerical statistics raised by Nash and Sutcliffe [1970], the quantitative uncertainty framework by Beven and Binley [1992], the discussion on model accuracy, uncertainty and realism by Wagener *et al.* [2001], and the diagnostic framework by Gupta *et al.* [2008], which is discussed below.

The most common and straightforward approach to evaluate hydrological models is to *quantitatively evaluate model behavior* by examining the difference between model output and observed variables. A large variety of statistical measures can be calculated from this difference that express the accuracy of a model. However, since there is uncertainty

in both measurements and model (discussed below), hydrologists have understood the value of providing the associated probability of model output. This information requires uncertainty in the data and model to be quantified and this has proven to be a great challenge for the hydrological community.

Qualitative evaluation of consistency in model behavior is another important aspect of model evaluation. Examples of this range from visual inspection of expected patterns in model output to model parameter sensitivity analysis. In recent years, several approaches have been suggested that combine quantitative and qualitative information into a more complete model evaluation framework. Examples are the GLUE method [Beven and Binley, 1992; Beven, 2006], the multi-criteria approach [Gupta *et al.*, 1998], and DYNIA [Wagener *et al.*, 2003b].

Finally, there is *qualitative evaluation of consistency in model form (structure) and function (behavior)*. This implies that certain model structures might be favored over others based on, for example, their performance on past problems or their similarity to the perceptual model (see Beven [2001b]) of the system at hand. Although subjective, this can be a valuable contribution in model evaluation.

The above three forms of model evaluation are based on the framework presented by Gupta *et al.* [2008], who argue that evaluation should be diagnostic in nature, i.e. focusing on identifying which components of the model could be improved and how. It is for this reason that model evaluation should focus not merely on a simple comparison of series of model output and observed data of that variable, but on comparing signature information through which the essence of model and data is extracted.

1.2.4 Issues of Uncertainty

All components of a hydrological modeling application are subject to uncertainty. The three primary sources of uncertainty, as discussed by Y. Q. Liu and Gupta [2007], are as follows.

1. Data — Observations of model input, states and output inherently contain measurement errors, which can be divided into instrument errors (i.e., imperfect measurement devices or procedures) and representativeness errors (i.e., incompatibility between observed and model variables, for instance in terms of scale).
2. Model parameters — Parameters are conceptual aggregate representations of spatially and temporally heterogeneous properties of the real system. They usually cannot be easily directly related to observable real-world characteristics of a catchment. Model calibration is usually applied to let model output comply to observations in an acceptable manner, as a result of which errors and uncertainties are introduced.

3. Model structure — Models are assemblies of assumptions and simplifications and thus inevitably imperfect approximations to the true system. If these approximations are inadequate, large errors can be the result. Mathematical implementation issues can also add to model structural uncertainty.

Future research not merely calls for objective, quantitative, and accurate estimations of model output uncertainty, but also for a minimization of this uncertainty. Suggestions for uncertainty estimation are widespread in recent literature (e.g., Beven and Binley [1992]; Thiemann *et al.* [2001]; Vrugt *et al.* [2003a, 2005]; Kuczera *et al.* [2006]). To accomplish the minimization of uncertainty, however, much more radical steps are needed that relate to the very fundamentals of hydrological modeling. Examples of this are acquiring new and better hydrological observations, finding improved methods of extracting and using information from observations, and improved hydrological models using better system representations and mathematical techniques.

1.3 Research Objectives and Outline

The main objective of this research is to apply CI techniques to catchment-scale R–R modeling in order to find improved methods of developing and evaluating such models. A clear focus of this research is on making better use of the information contained in both observations and model output. Three fields of application of CI techniques in R–R modeling are explored for these purposes. They are listed below, along with more specific objectives regarding their application.

1. Data-driven modeling — Find out if CI methods can model the R–R transformation adequately and how well they compare to conceptual hydrological models (see Chapters 3 and 5).
2. Parameter estimation — Apply CI parameter estimation algorithms to calibration of both CI and conceptual models to test whether more information can be extracted from hydrological data in order to make better R–R models (see Chapters 4 and 5).
3. Data mining — Find and make use of dynamical patterns in hydrological data that are commonly ignored in model evaluation (see Chapter 6).

Chapter 2 presents a short literature review of the three fields of application of CI mentioned above. In Chapter 3, a R–R model based on a well-known CI technique (an artificial neural network) is developed. Some important issues regarding the development, calibration and performance of such models are highlighted and discussed. Chapter 4 presents the application of evolutionary, multi-criteria algorithms to the calibration of artificial neural network R–R models, along with a comparison with traditional single-criterion algorithms.

A multi-criteria comparison of an artificial neural network model and a conceptual hydrological model is subsequently presented in Chapter 5. In Chapter 6, a temporal clustering approach was employed to identify periods of hydrological similarity. The results were used to shown how the evaluation of a conceptual model can be improved to be more diagnostic in nature and how subsequent improvements to the model structure can be inferred. The conclusions and recommendations of this thesis are presented in the seventh and final chapter.

Chapter 2

Computational Intelligence in Rainfall–Runoff Modeling

2.1 Fields of Application

CI methods have become — and are still becoming — increasingly common in R–R modeling. This chapter presents a brief overview of three typical applications of CI, all of which are tested in the remainder of this work. Firstly, hydrological system identification is discussed in Section 2.2. This application has a long history (see Dooge and O'Kane [2003]) but with the emergence of CI methods it is experiencing somewhat of a resurgence in the form of what is nowadays commonly termed data-driven modeling. Secondly, Section 2.3 discusses the application of parameter estimation methods from CI applied to the calibration of R–R models. Finally, in Section 2.4, some examples of data mining methods related to R–R modeling are discussed. Usually data mining methods do not apply directly to R–R problems but they can be used for pre-analysis of data or post-analysis of results.

2.2 Data-Driven Modeling

2.2.1 Knowledge-Driven versus Data-Driven

A common approach to simulate catchment systems is to model them based on process knowledge. This so-called knowledge-driven approach aims to represent the real-world hydrological system and its behavior in a physically realistic manner, and is therefore based on detailed descriptions of the system and the processes involved in producing runoff. The best examples of knowledge-driven modeling are so-called physically-based model approaches, which generally use a mathematical framework based on mass, momentum and energy conservation equations in a spatially distributed model domain, and parameter values that are directly related to catchment characteristics. These models require input of initial and boundary conditions since flow processes are described by differential equations. Examples of physically-based R–R modeling are the Système Hydrologique

Européen (SHE) [Abbott *et al.*, 1986a, 1986b] and the Representative Elementary Watershed (REW) [Reggiani *et al.*, 2000; Reggiani and Rientjes, 2005; Zhang and Savenije, 2005, 2006] model approaches. Physically-based modeling suffers from drawbacks due to the complexity of the R–R transformation process in combination with limitations in representing the small-scale spatial variability of meteorological inputs, physiographic characteristics, and initial conditions in the model (see Rientjes [2004]). Examples of drawbacks are excessive data requirements, large computational demands, and overparameterization effects. This is what causes modelers to look for more parsimonious and simple model approaches that incorporate a higher degree of empiricism, but it is (still) not clear how far this empirical approach should be taken (cf. Nash and Sutcliffe [1970] and Beven [2001a]). Conceptual model approaches are a first step from physically-based model approaches in a more empirical direction. These approaches use the principal of mass conservation in combination with simplified descriptions of the momentum and energy equations. Conceptual modeling commonly implies that the model domain is represented by storage elements, either in a spatially lumped or semi-distributed manner. Well-studied examples of conceptual modeling are the HBV model [Lindström *et al.*, 1997], the TOPMODEL [Beven *et al.*, 1995b], and the Sacramento soil moisture accounting model [Burnash, 1995]. Despite their popularity, there has been much debate in the literature on how much model complexity is warranted (e.g., Beven [1989]; Jakeman and Hornberger [1993]) and how their performance can be best evaluated (e.g., Klemeš [1986]; Gupta *et al.* [1998]).

The data-driven approach to R–R modeling, on the other hand, is based on extracting and re-using information that is implicitly contained in hydrological data without directly taking into account the physical laws that underlie the R–R processes (of which the principle of mass conservation is the most commonly implemented). The data-driven modeling paradigm is strongly related to the systems approach (see Section 1.2) and has been around since the very beginning of hydrological modeling. Basically, the first methods that tried to approximate the transformation from rainfall to runoff were empirical methods that relied on crude assumptions and subsequent fitting to data (see Beven [2001a] for a more complete historical perspective). Roughly since the beginning of the 1990s, interest in data-driven techniques has virtually exploded thanks to theoretical developments and an increase in available computational power. The field of data-driven modeling comprises a plethora of techniques, of which examples are discussed in Sections 2.2.3 and 2.2.4. Nowadays, traces of the data-driven paradigm can be found in many hydrological studies, but the full power of its techniques (many of which are still rapidly evolving) is likely not yet exploited, and insights into and experience with practical applications remain limited.

2.2.2 Advantages and Disadvantages

Data-driven R–R models are generally quickly and easily developed and implemented, and are less affected by the drawbacks of knowledge-driven models. Because of their

relative simplicity, simulation times often remain within reasonable limits. Moreover, their flexibility requires little expert knowledge of the system or processes modeled.

The latter argument could also be used against them, because naturally the reliance on data alone poses some difficulties:

- Because of their low transparency, which results from the inability to interpret their internal workings in a physically meaningful way, data-driven models generally fail to give useful insights into the system under investigation.
- Data inherently contains (measurement and scale) errors, which can translate to serious model deficiencies.
- How to ensure that a data-driven model learns the correct relationships from the data? Fitting a flexible model structure to data does not assure a reliable model.
- The range of application can be limited because empirical models only have validity over the range of the specific sample of the hydrological records that is used for model calibration. Extrapolation results beyond this range are therefore often inaccurate and uncertain. The same argument applies to situations in which a system has changed.

For these and other reasons, physical insights should be incorporated into the model development procedure where possible. Still, one might not be able to overcome the inherent flaws of data-driven models. For an insightful discussion on the shortcomings and risks of the data-driven paradigm, see the article by Cunge [2003].

2.2.3 Artificial Neural Networks as Data-Driven Models

A data-driven technique that has gained significant attention in recent years is Artificial Neural Network (ANN) modeling. In many fields, ANNs have proved to be good in simulating complex, non-linear systems, while generally requiring little computational effort. This awareness inspired hydrologists to carry out early experiments using ANNs in the first half of the 1990s. Their promising results led to the first studies on the specific topic of ANNs for R–R modeling (e.g., Halff *et al.* [1993]; Hjemfelt and Wang [1993]; Karunanithi *et al.* [1994]; Hsu *et al.* [1995]; Smith and Eli [1995]; Minns and Hall [1996]). ASCE [2000] and Dawson and Wilby [2001] give reviews on ANN modeling in hydrology. The majority of studies proved that ANNs are able to often outperform traditional statistical R–R techniques (e.g., Hsu *et al.* [1995]; Shamseldin [1997]; Sajikumar and Thandaveswara [1999]; Tokar and Johnson [1999]; Thirumalaiah and Deo [2000]; Toth *et al.* [2000]; A. Jain and Indurthy [2003]; Huang *et al.* [2004]) and to also produce good results compared to conceptual R–R models (e.g., Hsu *et al.* [1995]; Tokar and Markus [2000]; Dibike and Solomatine [2001]; de Vos and Rientjes [2007]). The field of R–R modeling using ANNs is nevertheless still in an early stage of development and remains a topic of continuing interest. Examples of ANN R–R modeling research in recent years include Minns [1998],

Campolo *et al.* [1999], Abrahart and See [2000], Gaume and Gosset [2003], Anctil *et al.* [2004], A. Jain and Srinivasulu [2004], Rajurkar *et al.* [2004], Ahmad and Simonovic [2005], de Vos and Rientjes [2005], A. Jain and Srinivasulu [2006], de Vos and Rientjes [2007], Han *et al.* [2007], Kamp and Savenije [2007], Srivastav *et al.* [2007], and de Vos and Rientjes [2008a]. Still, more research is needed to support the discussion on the value of these techniques in this field and to help realize their full potential, especially since their black-box nature, their flexibility and their automatic adjustment to information makes them prone to the risk of producing results that lack consistency or plausibility. Chapters 3, 4 and 5 of this thesis discuss ANNs and their application to R–R modeling in detail.

2.2.4 Other Data-Driven Model Techniques

Regression

A multiple linear regression model such as the example presented in Equation 2.1, can be seen as a simple example of a data-driven technique. Regression is a so-called parametric approach, meaning it requires *a priori* formulation of the form of the relationship between the dependent variable Z and the independent variable X. This form is usually linear, but regression can be extended to non-linear cases. The parameters of the model need to be calibrated to best fit the observed data Z given the noise signal ϵ.

$$Z = a_0 + a_1 X + \epsilon \tag{2.1}$$

Regression trees and model trees are variations on classical regression methods that consist of local regression models for separate parts of the complete data set. A hydrograph, for instance, can be classified into several categories, after which a separate regression model is built for each category (e.g. Solomatine and Dulal [2003]).

Time Series Modeling

Time series modeling is a linear data-driven technique, whose general framework is described by Box and Jenkins [1976]. Most time series contain an autoregressive (AR) component that accounts for the delay in the series, and a moving average (MA) component of a time series that is an expression of its memory. A difference term (I) can also be added to account for trends in the series. In case the time series under investigation has a clear correlation with another time series (e.g. like streamflow depends on rainfall), the latter can be used as an additional exogenous (X) variable in the model. An example formulation for such an ARIMAX model is as follows.

$$Z_t - Z_{t-1} = \mu + b_1 Z_{t-1} + b_2 Z_{t-2} + ... + w_1 X_{t-1} + w_2 X_{t-2} + ... + \epsilon_t - a_1 \epsilon_{t-1} - a_2 \epsilon_{t-2} \tag{2.2}$$

where μ is the average difference in Z_t, and b, w and a are parameters.

The application of time series models for the forecasting of streamflow has a long history (see W. Wang [2006]). Classical time series models like ARIMA, however, assume that the time series under study are generated from linear processes, which is generally not the case in hydrology. Some nonlinear regression-type time series models such as Autoregressive Conditional Heteroscedasticity (ARCH) models have been tested in streamflow modeling (e.g. W. Wang *et al.* [2005]).

Support Vector Machines

An increasingly popular technique from CI is the Support Vector Machine (SVM), developed by Vapnik [1998]. This nonlinear classification and regression technique has a strong similarity to the ANN, and is theoretically reliable in extracting relationships from data while ignoring noise. Examples of successful applications to R–R modeling include the works by Dibike [2002], Liong and Sivapragasam [2002], Bray and Han [2004], Asefa *et al.* [2002] and Chen and Yu [2007].

Fuzzy Methods

Fuzzy methods are based on a 'fuzzy' instead of the traditional 'crisp' representation of information. The idea is to express information as a degree of truth (not to be confused with uncertainty). Examples of application in R–R modeling can be found in, for instance, Bárdossy and Duckstein [1995], Nayak *et al.* [2005] and Vernieuwe *et al.* [2005].

Genetic Programming

Genetic programming is an evolutionary method that can be used for regression purposes. So far, only a few applications to R–R modeling have been reported in the literature (e.g. Khu *et al.* [2001]; Babovic and Keijzer [2002]).

Data-Based Mechanistic Modeling

In Data-Based Mechanistic (DBM) modeling [Young and Beven, 1994; Young, 2001, 2003], a model is built from a general class of model structures. DBM is an example of a stochastic, top-down approach to modeling which is more parsimonious than most data-driven techniques. Most importantly, DBM allows for a physical interpretation of the model. This approach can be seen as trying to strike a balance between data-driven and knowledge-driven modeling.

Table 2.1: The five major characteristics complicating the optimization in conceptual R–R modeling. Reproduced from Duan *et al.* [1992].

Characteristic	Reason for Complication
1. Regions of attraction	more than one main convergence region
2. Minor local optima	many small "pits" in each region
3. Roughness	rough response surface with discontinuous derivatives
4. Sensitivity	poor and varying sensitivity of response surface in region of optimum, and nonlinear parameter interaction
5. Shape	nonconvex response surface with long curved ridges

2.3 Parameter Estimation

2.3.1 Automatic Calibration Methods

Given the fact that hydrological models never perfectly represent the real world, the parameters of the model are fine-tuned in a calibration procedure to match the model output with observed data. The literature on this complex issue is vast, but a good overview of recent developments in hydrological model calibration is presented in the book of Duan *et al.* [2003]. Nowadays, modelers often use the capabilities and speed of digital computers by applying automatic optimization algorithms to find well-performing parameter values. Table 2.1 introduces the five main characteristics that were found by Duan *et al.* [1992] that complicate the automatic calibration of conceptual R–R models, the most important of which is considered to be the presence of many local optima.

Traditional optimization algorithms usually search by starting at a randomized or chosen point in the parameter space and following a single path to find an optimum. Such algorithms usually depend on local search mechanisms (e.g. following the gradient of the response surface) and therefore run the risk of getting stuck in local optima or failing because of any of the other characteristics mentioned in the table. Global optimization algorithms have been developed in recent years that are claimed to search the parameter space more extensively and efficiently. Most CI optimization algorithms have global optimization capabilities and are therefore promising in dealing with the problems related to R–R model calibration. The following subsections address the two main families of such algorithms.

2.3.2 Evolutionary Algorithms

Evolutionary algorithms are inspired by Darwin's theory of evolution. The main idea is to evolve a population of possible solutions to a given problem by applying principles of natural selection:

- Selection – only the 'fittest' members of a population are copied into the next generation
- Crossover – members produce offspring by exchanging characteristics
- Mutation – a population member will occasionally randomly mutate (some of) its characteristics

The population members are usually different models or model parameter sets, and the fitness is expressed by the difference between model output and observations. The algorithms use the rules mentioned above, and have their population size, the number of generations, and the probabilities of crossover and mutation as most important settings. Evolutionary methods have been shown to elegantly find globally optimal solutions to many problems.

The most common evolutionary algorithms is the Genetic Algorithm (GA), introduced by Holland [1975]. Many different implementations of the natural selection rules mentioned above have been suggested for the GA. In Chapter 4 of this work, a traditional form of the GA has been used to calibrate an ANN R–R model. A popular example of an evolutionary algorithm developed in hydrology is the Shuffled Complex Evolution algorithm developed by Duan *et al.* [1992]. A more recently introduced evolutionary algorithm is the so-called Differential Evolution (DE) algorithm by Storn and Price [1997]. In this work, a variation of DE is applied for the calibration of a conceptual R–R model. The results, along with a detailed explanation of DE's working are presented in Chapter 6.

In recent years, evolutionary algorithms have been applied frequently to a plethora of optimization problems including R–R modeling. Well-known examples in R–R model calibration include Q. J. Wang [1991], Duan *et al.* [1992], and Franchini and Galeati [1997].

2.3.3 Biologically-Inspired Algorithms

A new and developing field in CI is the use of techniques inspired by the behavior of groups of animals (e.g., the flocking of birds or schooling of fish). Typically, a population of simple agents is modeled that are allowed to interact amongst themselves and with their environment. This can lead to the emergence of global behavioral patterns, which is often referred to as swarm intelligence. Groups of animals make use of this intelligence, for example in their search for food. Inspired by this, researchers have developed swarm intelligence techniques that search for optima on the response surface of functions. A well-known example of such an optimization technique is Particle Swarm Optimization [Kennedy and Eberhart, 1995]. Another example of a biologically-inspired algorithm is

Ant Colony Optimization [Dorigo and Stützle, 2004], which searches the parameter space in the way ants navigate using pheromone trails. Several successful applications of swarm intelligence techniques in R–R modeling have been presented (e.g. Chau [2006]; Kashif Gill *et al.* [2006]; Goswami and O'Connor [2007]).

2.3.4 Multi-Criteria Algorithms

A recent development in optimization is the development and application of multi-criteria (MC) algorithms that are able to simultaneously optimize multiple criteria and present the full range of trade-offs between them (the so-called Pareto front). The structure of evolutionary algorithms allows them to be easily translated into effective forms for MC optimization, and a number of MC evolutionary algorithms have been developed over recent years (e.g., SPEA [Zitzler and Thiele, 1999], MOSCEM–UA [Vrugt *et al.*, 2003a], NSGA–II [Deb, 2001], MOPSO [Kashif Gill *et al.*, 2006], and AMALGAM [Vrugt and Robinson, 2007]). As a result, recent literature shows an increasing number of studies that use MC algorithms for R–R model calibration (e.g., Gupta *et al.* [1998]; Yapo *et al.* [1998]; Boyle [2000]; Boyle *et al.* [2000]; Vrugt *et al.* [2003a]; Khu and Madsen [2005]; Kashif Gill *et al.* [2006]; Tang *et al.* [2006]; Fenicia *et al.* [2007a]; de Vos and Rientjes [2007, 2008b]). More discussion on MC theory and methods, along with applications and results, can be found in Chapters 4 and 5.

2.4 Data Mining

2.4.1 Data Mining and Cluster Analysis

Data mining techniques are tools to facilitate the conversion of data into forms that provide a better understanding of processes that generated or produced these data [Babovic, 2005]. A variety of techniques can be used, but among the most common are clustering algorithms. These algorithms automatically categorize information by finding clusters in the data. This could be useful, for example when trying to find homogeneous parts of the data, compressing the information in the data into a small number of discrete values, or finding relationships in the data that were not foreseen.

2.4.2 Clustering Algorithms

One of the simplest examples of a cluster algorithm is the k-means algorithm. It is discussed in more details in Chapter 6, where it is used for temporal clustering of hydrological data. It has also been used in R–R modeling for, for example, regionalization [Bhaskar and O'Connor, 1989; Burn, 1989; Mazvimavi, 2003], and identification of spatial clusters with similar seasonal flood regimes [Lecce, 2000].

Related to k-means clustering is its fuzzy c-means clustering variant, which uses the principle of fuzzy information to determine overlapping clusters. Choi and Beven [2007] use it for finding periods of hydrological similarity and subsequent model conditioning, and Xiong *et al.* [2001] for combining model outputs, for example.

Another interesting clustering algorithm is the Self-Organizing Map (SOM). It can be used for clustering but is also a unique method for visualizing information in data. The technique was used for clustering of watershed conditions by Liong *et al.* [2000], determination of hydrological homogeneous regions by Hall and Minns [1999], for finding periods of hydrological similarity and subsequent local modeling by Hsu *et al.* [2002], and model evaluation and identification by Herbst and Casper [2007].

Chapter 3

Constraints of Artificial Neural Networks for Rainfall–Runoff modeling

Modified from:

de Vos, N. J., Rientjes, T. H. M., 2005. Constraints of artificial neural networks for rainfall–runoff modeling: trade-offs in hydrological state representation and model evaluation. *Hydrol. Earth Sys. Sci.* **9**, 111–126.

and

de Vos, N. J., Rientjes, T. H. M., 2008. Correction of timing errors of artificial neural network rainfall-runoff models. In: Practical Hydroinformatics, Abrahart, R. J., See, L. M., Solomatine, D. P. (eds.)., Springer Water Science and Technology Library.

Abstract

The application of ANNs in R–R modeling needs to be researched more extensively in order to appreciate and fulfill the potential of this modeling approach. This chapter reports on the application of multi-layer feedforward ANNs for R–R modeling of the Geer catchment (Belgium) using both daily and hourly data. The daily forecast results indicate that ANNs can be considered good alternatives for traditional R–R modeling approaches, but the simulations based on hourly data reveal timing errors as a result of a dominating autoregressive component. This component is introduced by using previously observed runoff values as ANN model input, which is a popular method for indirectly representing the hydrological state of a catchment. Two possible solutions to this problem of lagged predictions are presented. Firstly, several alternatives for representation of the hydrological state are tested as ANN inputs: moving averages over time of observed discharges and rainfall, and the output of the simple GR4J model component for soil moisture. A combination of these hydrological state representations produces good results in terms of timing, but the overall goodness of fit is not as good as the simulations with previous runoff data. Secondly, an aggregated objective function was tested that penalizes the ANN model for having a timing error. The gradient-based training algorithm that was used had difficulty

with finding good optima for this function, but nevertheless some promising results were found. There seems to be a trade-off between having good overall fit and having correct timing, so further research is suggested to find ANN models that satisfy both objectives.

3.1 Introduction

As discussed in Chapter 1, one of the main research challenges in hydrology is the development of models that are able to accurately simulate a catchment's response to rainfall. Such models are capable of forecasting future river discharge values, which are needed for hydrological and hydraulic engineering design and water management purposes. However, simulating the real-world relationships using these R–R models is far from a trivial task since the various interacting processes that involve the transformation of rainfall into discharge are complex, and they vary in space and time. Hydrologists have attempted to address this modeling issue from two different points of view: using knowledge-driven modeling and data-driven modeling (see Section 2.2).

In order to investigate the evolving field of ANN R–R modeling, several ANN design aspects are investigated through a case study in this chapter. Multi-layer feedforward ANN models are developed for forecasting short-term streamflow, and both hourly and daily data sets from the Geer catchment (see Appendix A) are used to develop and to test the ANN models. Particular attention is paid to the representation of the hydrological state (i.e., the amount and distribution of water storage in a catchment) in ANN models. Since the hydrological state greatly determines a catchment's response to a rainfall event, it is a critical model input. Previous discharge values are often used as ANN inputs, since these are indirectly indicative for the hydrological conditions. In this chapter, the negative consequences of this approach are discussed and several alternatives for state representation are tested. Moreover, the shortcomings of current evaluation methods of ANN models in the calibration phase are discussed.

3.2 Artificial Neural Networks

3.2.1 Introduction

ANNs are mathematical models that consist of simple, densely interconnected elements known as neurons, which are typically arranged in layers (see Fig. 3.1). An ANN receives signals through input units and these signals are propagated and transformed through the network toward the output neuron(s). In this study, so-called feedforward ANNs are used, in which information always travels in the direction of the ANN output without delay. One of the key transformations performed by an ANN is multiplication with weights

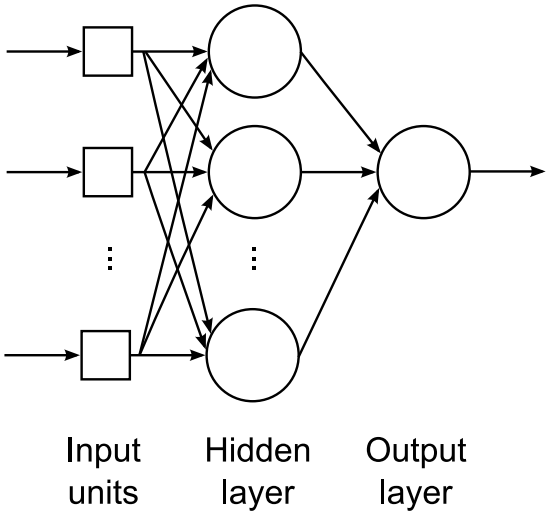


Figure 3.1: An exemplary feedforward ANN with one hidden layer. The input units are not considered neurons since they do not transform data and merely pass information to the network.

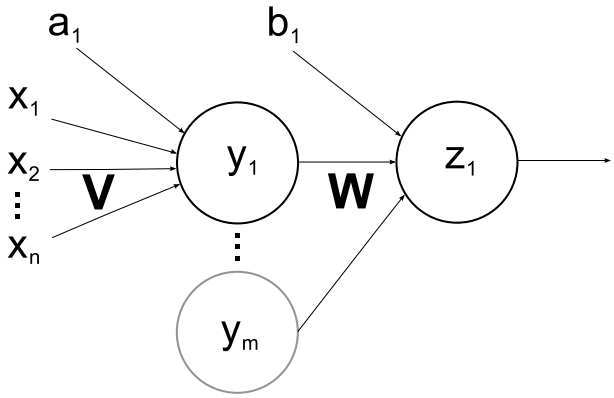


Figure 3.2: Schematic representation of the structure of an ANN.

that express the strength of connections between neurons. During a calibration procedure known as training, the internal pattern of connectivity between neurons — i.e., the weights, and therefore the model's response — is adapted to information that is presented to the network. Section 3.2.2 addresses this training procedure in more detail.

A typical feedforward ANN with one hidden layer can be mathematically expressed by the following set of equations:

$$\begin{aligned} z_k &= \sum_{j=1}^{m} y_j w_{jk} + b_k \\ y_j &= f\left(\sum_{i=1}^{n} x_i \dot{v}_{ij} + a_j\right) \end{aligned} \tag{3.1}$$

where v_{ij} and w_{jk} are weights and a_j and b_k the biases for the hidden and output neurons, respectively. The function f is a so-called transfer function, for which a sigmoid function is commonly chosen.

Figure 3.2 presents the above in a structural view of an ANN with input vector $\mathbf{x}$, one hidden layer with an activation value vector $\mathbf{y}$, and one output z_1. The matrices $\mathbf{V}$ and $\mathbf{W}$ are the weight matrices for the connections between the two layers.

Background information about the wide array of ANN techniques and details about their workings can be found in many excellent textbooks such as Hecht-Nielsen [1990] and Haykin [1999].

3.2.2 Training and Evaluation

ANNs are commonly trained by an optimization algorithm, which attempts to reduce the error in network output by adjusting the network weights and the neuron biases. The common approach to ANN training in function approximation applications such as R–R modeling is to use supervised training algorithms. These algorithms are used in combination with sample input and output data of the system that is to be simulated. The weights are changed according to the optimization of some performance measure, which is a measure of the degree of fit (or difference) between the network estimates and the sample output values. The alteration of network weights in the training phase is commonly stopped before the training optimum is found, because then the network is supposed to have also learned the noise in the training data and to have lost its generalization capability. This situation is referred to as overtraining of an ANN. Undertraining, on the other hand, occurs when the training is stopped too early for the ANN to learn all the information contained in the training data. Both situations are likely to result in sub-optimal operational performance of an ANN model. It is for this reason that the available data are commonly split in three separate data sets: (1) the training set, (2) the cross-evaluation set, and (3) the evaluation set. The first provides the data on which an ANN is trained. The second is used during the training phase to reduce the chance of overtraining of the network. The minimization of the training error is stopped as soon as the cross-evaluation error starts to increase. This point is considered to lie between the undertraining and overtraining stages of an ANN, since a rise of the cross-evaluation error indicates that the ANN loses its capability to generalize from the training data. The latter of the three data sets is used to validate the performance of a trained ANN. This so-called split-sampling method is also applied in this study, and all results are presented for the evaluation data set.

The measures for evaluating model performance that are used in this chapter are the Root Mean Square Error (RMSE), the Mean Squared Logarithmic Error (MSLE) [Hogue *et al.*, 2000], the Nash–Sutcliffe coefficient of efficiency, C_{NS} [Nash and Sutcliffe, 1970], and the Persistence Index, C_{PI} [Kitanidis and Bras, 1980]. They are defined as follows.

$$MSLE = \frac{1}{K} \sum_{k=1}^{K} \left(\ln \hat{Q}_k - \ln Q_k \right)^2 \tag{3.2}$$

$$C_{NS} = 1 - \frac{F}{F_0} \tag{3.3}$$

where

$$F = \sum_{k=1}^{K} \left(\hat{Q}_k - Q_k\right)^2 \tag{3.4}$$

$$F_0 = \sum_{k=1}^{K} \left(Q_k - \bar{Q}\right)^2 \tag{3.5}$$

$$C_{PI} = 1 - \frac{F}{F_p} \tag{3.6}$$

where

$$F_p = \sum_{k=1}^{K} \left(Q_k - Q_{k-1}\right)^2 \tag{3.7}$$

In these equations, K is the total number of data elements, Q_k and $\hat{Q}_k$ are the observed and the computed runoffs at the k^{th} time interval respectively, $\bar{Q}$ is the mean value of the runoff over time. F_0 is the initial variance for the discharge time series, F is the residual model variance, and F_p is the variance of the persistence model. The difference between the C_{NS} and C_{PI} is that the scaling of F for the latter involves the last known discharge value instead of the mean flow. This basically means that the model variance is compared with the variance of a model that takes the last observation as a prediction. C_{NS} and C_{PI} values of 1 indicate perfect fits. The RMSE is indicative for high flow errors and the MSLE for low flow errors.

At the start of each training trial, ANN weights and biases have to be initialized. The most-often applied method is random initialization. The goal of this randomization is to force the training algorithm to search various parts of the parameter space, thereby enabling a more robust overall optimization procedure and increasing the overall chances of finding a global error minimum. A result of this approach is that the performance of an ANN is often different for each training trial, even if it is trained using the same algorithm. There are three reasons why training algorithms do not find the same parameter set for each training trial when training starts in a different part of the parameter space. First of all, there may be more than one region of attraction for the training set to which the model can converge. Secondly, a training algorithm may not be able to find a global optimum and get stuck in local optima, on flat areas or in ridges on the response surface. Lastly, in case of applying cross-evaluation to prevent overtraining, the optimum in terms of the training data will probably not coincide with the optimum for the cross-evaluation

set. Therefore, an algorithm might be stopped before finding a global optimum due to increasing cross-evaluation errors. In the case of random initialization, the performance of an ensemble of training trials yields information on the parameter uncertainty of an ANN model type in combination with a certain training algorithm. Presenting this uncertainty allows for a more reliable and accurate comparison between combinations of ANN model types and training algorithms. Performing and presenting only a single training trial would be based on the assumption that a single trial represents a reliable indicator for the average performance, but experience has learned that this assumption is risky since ANN performance can vary considerably between training trials. Gaume and Gosset [2003], aware of this issue, addressed it by presenting ANN performance using Box-and-Whisker plots of the RMSE over an ensemble of 20 training trials. In this study, the mean and standard deviations of the performance measures over an ensemble of 10 training trials are presented. This ensemble size was found to be appropriate for quantifying parameter uncertainty of the ANN models while keeping calculation times acceptable. Time series plots and scatter plots are presented for the median of the ensemble.

3.2.3 Advantages and Disadvantages

ANNs have advantages over many other techniques since they are able to simulate non-linearity in a system. They can also effectively distinguish relevant from irrelevant data characteristics. Moreover, they are nonparametric techniques, which means that ANN models do not necessarily require the assumption or enforcement of constraints or *a priori* solution structures [French *et al.*, 1992]. This, in combination with the fact that ANNs are commonly automatically trained, makes that relatively little knowledge of the problem under consideration is needed for applying them successfully. Lastly, because of their compact and flexible model structure, ANNs have relatively low computational demands and can easily be integrated with other techniques.

A disadvantage of ANNs, however, is that the optimal form or value of most network design settings (such as the number of neurons in the hidden layer) can differ for each application and cannot be theoretically defined, which is why they are commonly determined using trial-and-error approaches [Zijderveld, 2003]. Another important drawback is that the training of the network weights tends to be problematic, which is due to the following reasons: (1) optimization algorithms are often unable to find global optima in complex and high-dimensional parameter spaces, (2) overparameterization effects may occur, and (3) error minimization in the training phase does not necessarily imply good operational performance. The latter pertains to the representativeness of the training data for the operational phase. For example, the training data should ideally reflect the distribution of variables in the operational situation, and should not contain many errors.

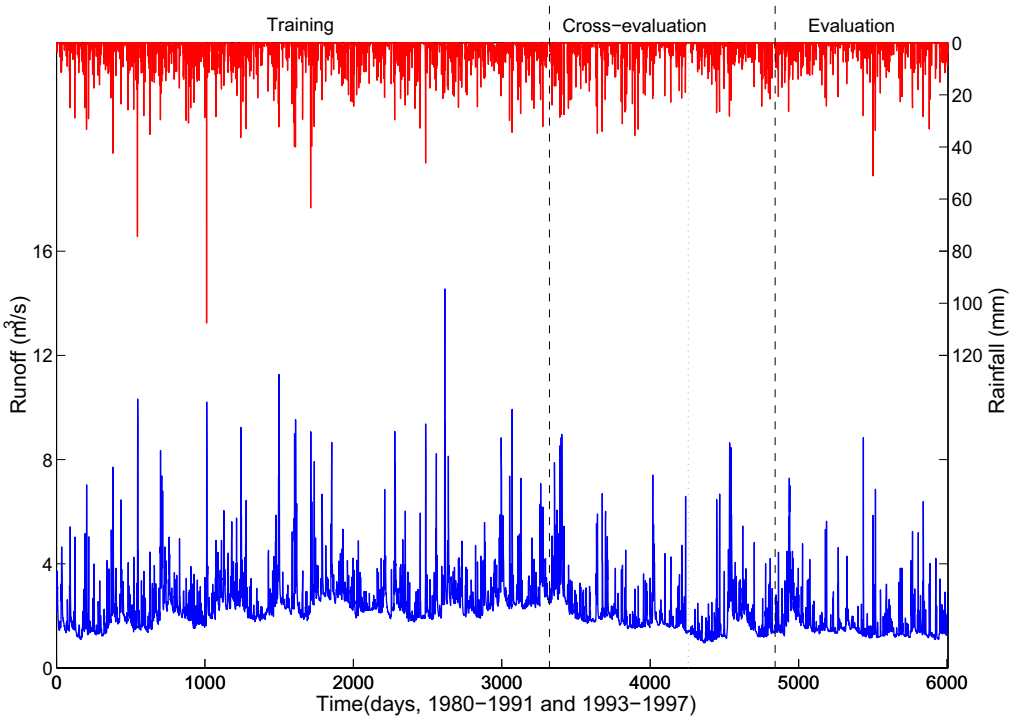


Figure 3.3: Daily runoff (Kanne) and rainfall (Bierset) from 1980 to 1991 and 1993 to 1997.

3.3 Case Study

3.3.1 Selected Data

Data from the Geer River basin were used for this research (see Appendix A). Figure 3.3 shows the daily catchment discharge in combination with rainfall at location Bierset for the period 1980–1997 (without year 1992). Figure 3.4 shows the hourly data for the period 1993–1997. Both the daily and hourly time series were divided into 55% for training, 25% for cross-evaluation and 20% for evaluation (also depicted in 3.3 and 3.4). All three fragments of the time series start with a period of constant low discharge and rainfall. The shapes of the discharge distributions over the three separate periods are similar for both the daily and the hourly data.

3.3.2 Input Signals

The ANN type that is used for R–R modeling is the static multi-layer feedforward network (see Figure 3.1). Static networks do not have the dimension of time incorporated in the network architecture, as opposed to dynamic networks, which use feedback connections or local memories in neurons. These static ANNs are nevertheless able to capture the dynamics of a system in the network model by using so-called tapped delay lines. This method presents a sequence of time series values (e.g., $P(t), P(t-1), \ldots, P(t-m)$) as

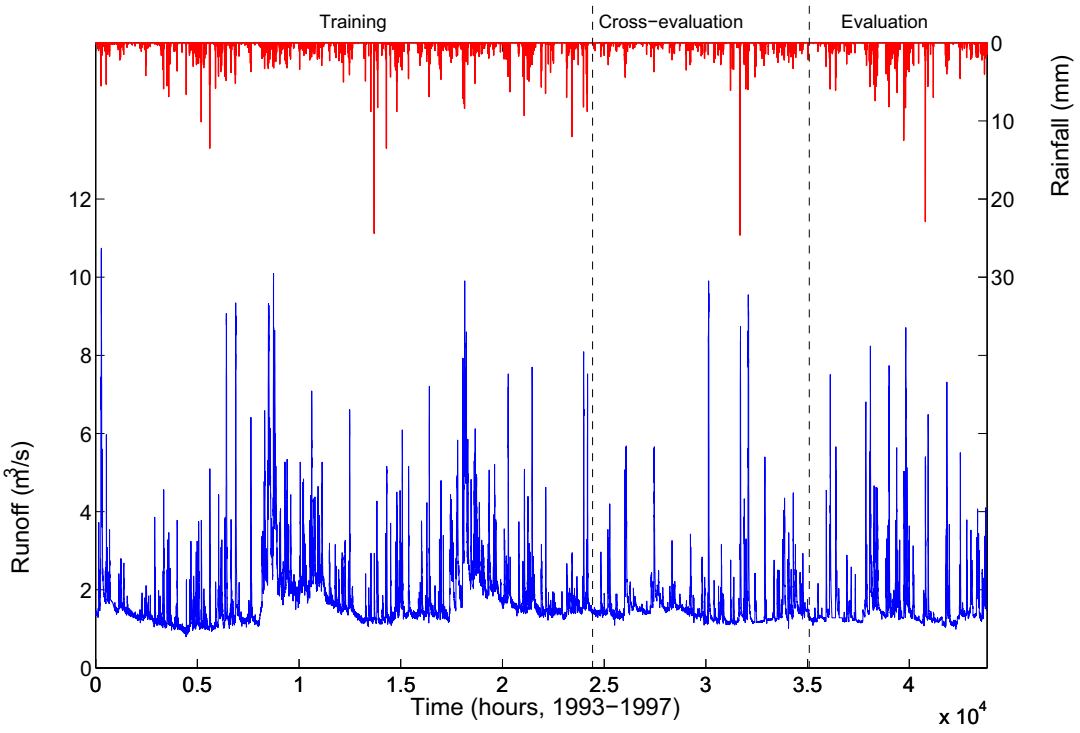


Figure 3.4: Hourly runoff (Kanne) and rainfall (Bierset) from 1993 to 1997.

separate network input signals. $P(t)$ represents an input variable in time and m the size of the time window. The number of input units thus increases with the size of this window.

The input signals to an ANN model should comprise all relevant information on the target output, and on the other hand, they should contain as little irrelevant information as possible. However, in order to facilitate the training procedure, largely overlapping information content of input signals should be avoided. Because an increased number of input signals leads to a more complex network structure, the task of training algorithms is being complicated, which is likely to have a negative effect on network performance. In order to make a parsimonious selection of ANN inputs, the popular approach of examining the linear correlations between the input and output time series was followed. Note that a non-linear technique such as an ANN, however, might be able to make use of more information than is revealed by this linear technique. Figures 3.5 and 3.6 show the correlation coefficients for various time lags between the time series of several observed variables and the daily and hourly time series of runoff at Kanne. The autocorrelation of the discharge time series is also presented. The minimum and maximum delays were chosen in such a way as to enclose high values of the correlation for each variable, thereby ensuring a high information content for each of the input signals. The catchment mean lag time is around 8 hours, which can be concluded from the peak correlation between the discharge time series and the rainfall series.

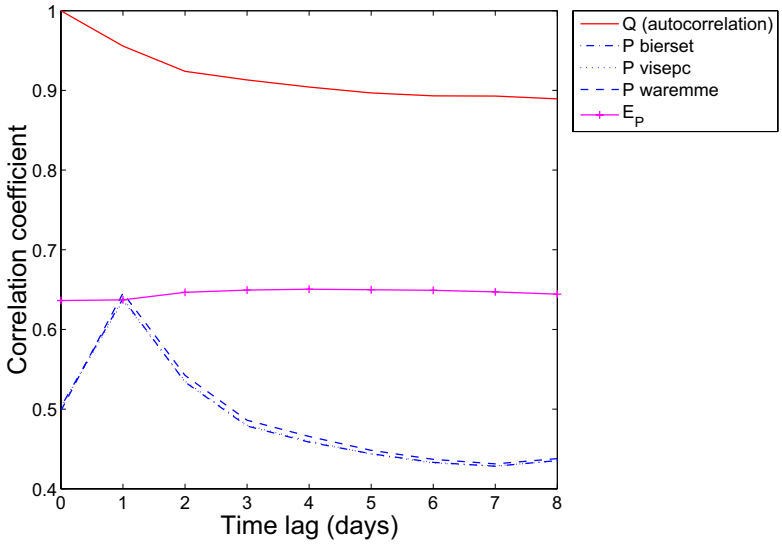


Figure 3.5: Correlation between the daily runoff time series and various other time series (rainfall, potential evaporation) for various time lags.

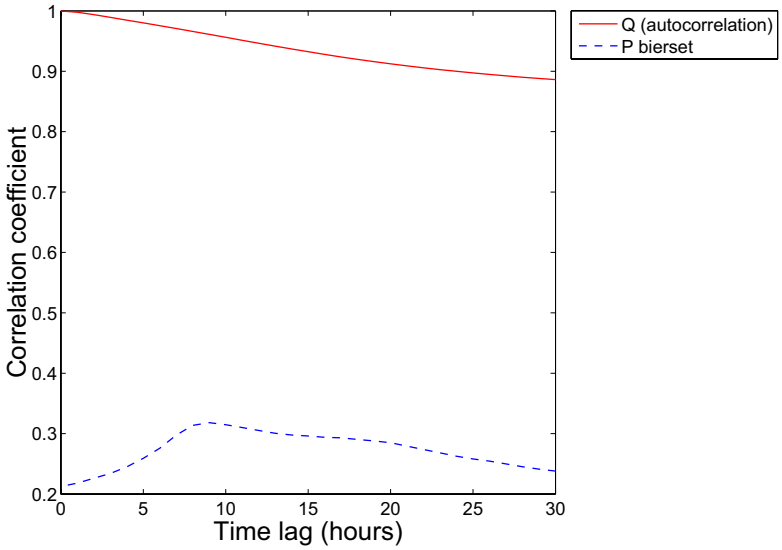


Figure 3.6: Correlation between the hourly runoff time series and the rainfall time series for various time lags.

Because the transfer functions that were used in this study become saturated outside a certain range, all input data are linearly scaled to a range of -1 to 1. The output of this transfer function is bounded to the range of -1 to 1, which is why the output data was scaled to a range of -0.8 to 0.7. The reason for setting these ranges narrow is to enable the ANN to extrapolate beyond the training data range, since extrapolation can be an important issue in the application of empirical methods such as ANNs. The output data range is asymmetrical because it is more likely that the upper bound of the training data range is exceeded than the lower bound. Even though previous research has shown that this approach to the problem of extrapolation has limitations (e.g., Minns [1998]), these measures will at least reduce the effects of the extrapolation problem where needed. Moreover, no extrapolation issues are expected since the training periods of both the daily and the hourly data contain the highest discharge values.

3.3.3 Training Algorithms

All ANNs were trained using supervised training algorithms that tried to minimize the Mean Squared Error (MSE) objective function. The merits of using a good algorithm are threefold: (1) better accuracy leads to better ANN performance, (2) faster convergence leads to smaller calculation times, and (3) lower spread in the performance makes it easier and more honest to evaluate and compare ANNs. Unfortunately, few algorithms are able to combine these three merits. The most well-known training algorithm is the backpropagation algorithm [Rumelhart and McLelland, 1986] that follows a steepest-descent approach based on the first-order gradient of the response surface. Other popular methods include the conjugate gradient algorithm [Fletcher and Reeves, 1964; Møller, 1993] and methods based on second-order gradients such as the Levenberg–Marquardt (LM) algorithm (see Hagan and Menhaj [1994]). Alternatives which were not tested here are the LLSSIM algorithm [Hsu *et al.*, 1995] and algorithms based on global optimization such as simulated annealing [Kirkpatrick *et al.*, 1983] and GAs [Goldberg, 2000].

The algorithms that were tested in this research are the steepest descent backpropagation (SD), steepest-descent with variable learning rate and momentum (SDvm), resilient steepest-descent backpropagation (RSD), Polak–Ribière, Fletcher–Reeves, and Powell–Beale conjugate gradient (CG–P, CG–F, CG–B), Broyden–Fletcher–Goldfarb–Shannon (BFGS), and LM algorithms. A so-called batch training approach was used for training the ANNs: the whole training data set is presented once, after which the weights and biases are updated according to the average error. Table 3.1 shows the one-step-ahead forecast performance in terms of the mean RMSE and C_{NS}, along with the number of iterations of the various algorithms. The latter gives an indication of the convergence speed of the algorithm. These indicative simulations were made with ANN models that are typical for this application and very similar to the ones used later in this study.

Table 3.1: Indication of ANN model performance using various training algorithms

	Daily data			Hourly data		
Algorithm	RMSE	C_{NS}	Iterations	RMSE	C_{NS}	Iterations
SD	1.275	-1.868	1000	0.572	0.411	800
SDvm	0.926	-0.568	140	0.948	-0.502	20
RSD	0.690	0.223	30	0.279	0.871	80
CG–P	0.770	0.010	25	0.206	0.929	60
CG–F	0.519	0.519	60	0.185	0.941	80
CG–B	0.425	0.706	50	0.164	0.956	90
BFGS	0.567	0.427	30	0.182	0.942	100
LM	0.339	0.815	20	0.151	0.963	40

The LM algorithm outperformed the other algorithms in terms of accuracy and convergence speed in all test cases. Moreover, the standard deviation of the LM algorithm was very low: 0.012 for daily data and 0.001 for hourly data. The other algorithms show much more spread in their performance measures (around 5 to 50 times more, depending on the algorithm), indicating that the LM algorithm may be considered much more robust.

The above results show that ANN model performance can be very dependent on the ability of optimization algorithms to find a good set of weights and biases, as also pointed out by, for example, Hsu *et al.* [1995]. However, many studies on ANN R–R models have relied on training algorithms such as the classic steepest-descent backpropagation algorithm, variants of that with momentum and/or variable learning rate, or conjugate gradient-based algorithms (see review by Dawson and Wilby [2001]). These results suggest that studies using multi-layer feedforward ANNs for R–R modeling would benefit from using more sophisticated algorithms such as LM.

3.3.4 Model Structure

Increasing the number of weights of an ANN by adding hidden neurons or layers, complicates network training. ANNs with one hidden layer are commonly used in rainfall-runoff modeling (see review by Dawson and Wilby [2001]) since these networks are considered to offer enough complexity to accurately simulate the dynamic and non-linear properties of the rainfall-runoff transformation. Preliminary test results showed that such ANNs indeed outperform the networks with two hidden layers. The optimal size of the hidden layer was found by systematically increasing the number of hidden neurons until the network performance on the test set no longer improved significantly. Figure 3.7 shows the performance of various ANN architectures in terms of the Nash–Sutcliffe coefficient. The ANN input for these simulations consisted of daily data with a total of 13 signals, concerning potential evaporation at one station, rainfall at three stations, and previous discharges. The

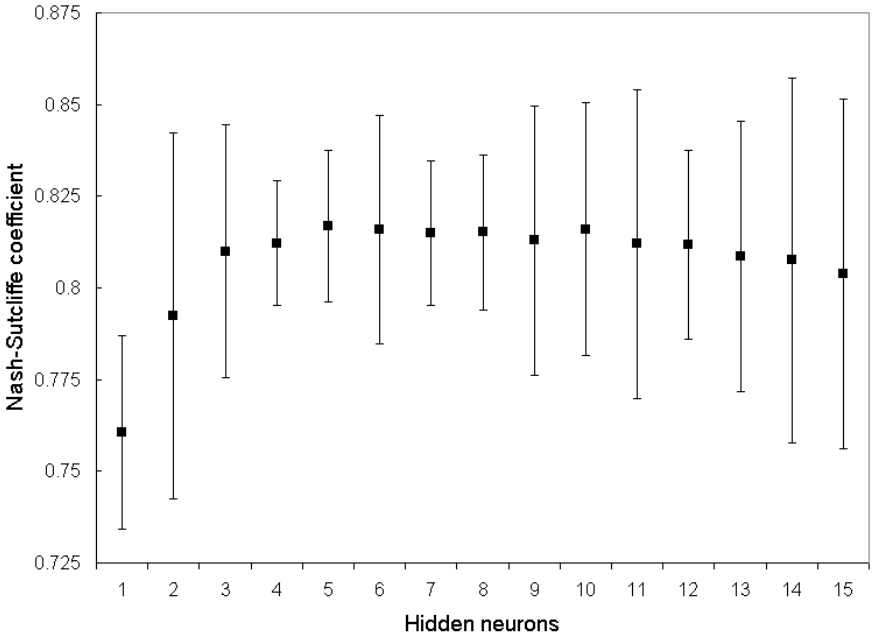


Figure 3.7: ANN performance (13 inputs, 1 output) for various hidden layer sizes. The squares represent the mean Nash–Sutcliffe coefficients C_{NS}, and the bars depict the 95% confidence bounds.

LM algorithm was used for training. The results show that there is a point at which the performance no longer increases (5 hidden neurons). Note that the 95% confidence bounds widen as the number of hidden neurons increases. This implies that the training algorithm is less likely to find optima as the dimensionality of the parameter space increases. Based on extensive testing, it was found that the optimal number of hidden neurons should be usually roughly around the square root of the number of input neurons.

ANN architectures with one output neuron were used throughout this study. The output signal from this neuron was the discharge prediction for a certain lead time. In order to make multi-step-ahead predictions (i.e., predictions with a lead time larger than one time step), two methods were available: (1) re-inputting a one-step-ahead prediction into the network, after which it predicts the two-step-ahead prediction, and so forth, and (2) by directly outputting the multi-step-ahead prediction. The first method uses the ANN's own preliminary estimations as a source of information for further predictions, the latter uses only the original data. Test results showed that for both the daily and hourly data the two methods performed nearly similar up to a lead time of respectively 4 days and 12 hours. Because of its simplicity, the direct multi-step-ahead method was used. Sigmoid-type functions are commonly used as transfer functions in hidden layers. The popular hyperbolic tangent function ($y = \tanh(x)$) was chosen here. The identity function ($y = x$) was used as transfer function in the output neuron.

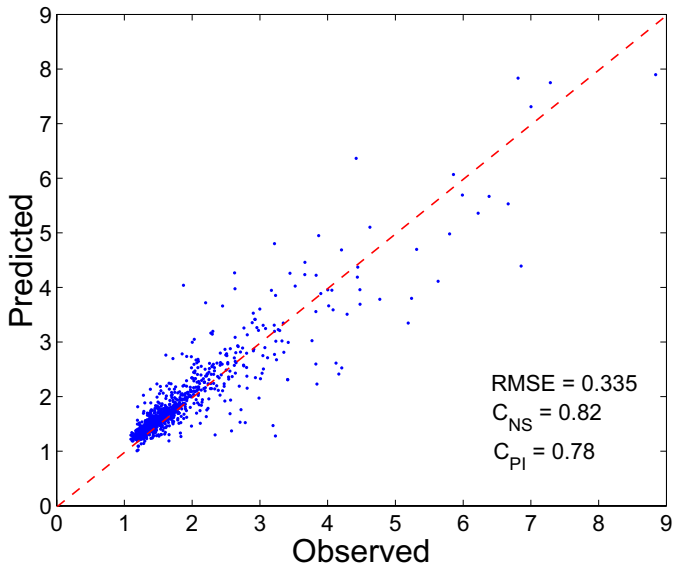


Figure 3.8: Scatter plot of predicted versus observed daily discharges (m^3/s) for a one-day-ahead forecast.

3.4 Results

3.4.1 Main Modeling Results

Figure 3.8 shows a scatter plot of the results of a one-day-ahead $(t+1)$ prediction of an ANN model using the daily data from the Geer catchment. The input to the network consisted of previously observed rainfall values at time instants t to $t-2$ at the three available measurement stations, potential evaporation at $t-3$, and discharge values at the catchment outlet from t to $t-2$. The ANN architecture was: $13-5-1$ (13 input units, 5 hidden neurons, 1 output neuron). A detail of the observed and predicted time series of the daily data is presented in Fig. 3.9. The ANN model proves to be able to make one-step-ahead forecasts with good accuracy, considering the forecast lead time is 24 hours and the catchment mean lag time merely 8 hours (see Fig. 3.6). The biggest drawback is that the model underestimates quite a number of moderate peak flows by up to 40%. However, Fig. 3.9 also shows that the model's timing of the peaks is quite good. Low flows are mostly well simulated, even though these forecasts show more fluctuations than the observed flow pattern. This is most likely due to the model overestimating the effect of small rainfall events.

Scatter plots of simulation results based on hourly data are presented in Figs. 3.10 and 3.11. The first shows the results of a one-hour-ahead discharge forecast using an $18-5-1$ ANN model with rainfall inputs from $t-5$ to $t-19$ and discharges from t to $t-2$. The latter presents the results of a six-hour-ahead forecast using a similar model (only the time window of the rainfall is shifted to t to $t-14$). Fig. 3.12 shows the mean and 95% confidence bounds of C_{NS} for increasing lead times. These results show that the ANN

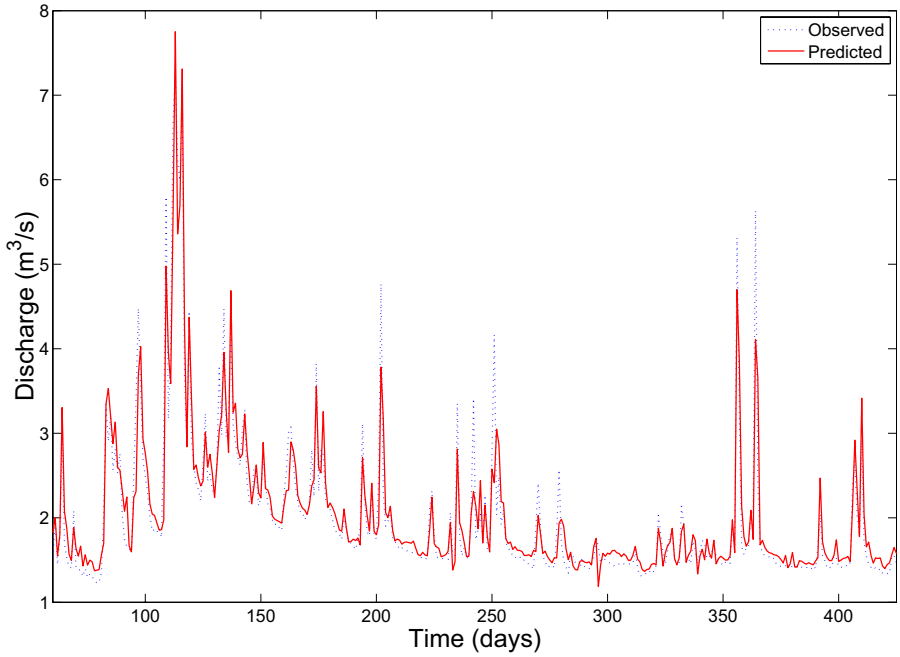


Figure 3.9: Observed and predicted daily time series of discharge (m^3/s) for a one-day-ahead forecast (detail; December 1, 1993 to December 1, 1994).

models are able to make good forecasts (in terms of the C_{NS}) for short lead times, but the performance decreases with increasing lead times. When forecasting 9 or more hours ahead, the performance deteriorates even more rapidly. This is due to the fact that rainfall up to time t, which are used as input signals, no longer contains significant information on the forecasted discharge, because the catchment's mean lag time is exceeded (cf. Fig. 3.6).

The scatter plot with low spread (Fig. 3.10), and the low RMSE and high C_{NS} of the one-hour-ahead forecast indicate excellent model performance, but the C_{PI} does not (also see Table 3.2). Moreover, the forecasts with longer lead times are not satisfactory, especially when compared with the forecast based on daily data. A visual interpretation of the simulation results, a representative detail of which is presented in Fig. 3.13, shows why: the prediction of the ANN model is lagged in comparison with the observed time series. This prediction lag effect is the result of using previously observed discharge values as ANN inputs. The high autocorrelation of the hourly discharge time series causes the autoregressive model component, which is implicitly contained in ANN models that use previously observed discharge values, to become dominant. The ANNs give the most weight to the latest discharge input (usually, Q at t) for calculating the forecast (Q at $t+L$). In other words, the ANN models say that the best forecast for the discharge over a certain lead time is close to the value of the currently observed discharge. In terms of most

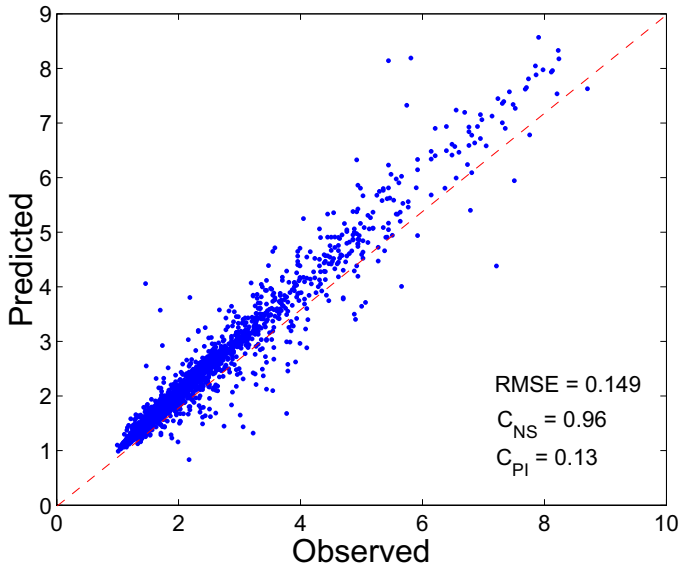


Figure 3.10: Scatter plot of predicted versus observed hourly discharges (m^3/s) for a one-hour-ahead forecast based on historical rainfall and discharge values.

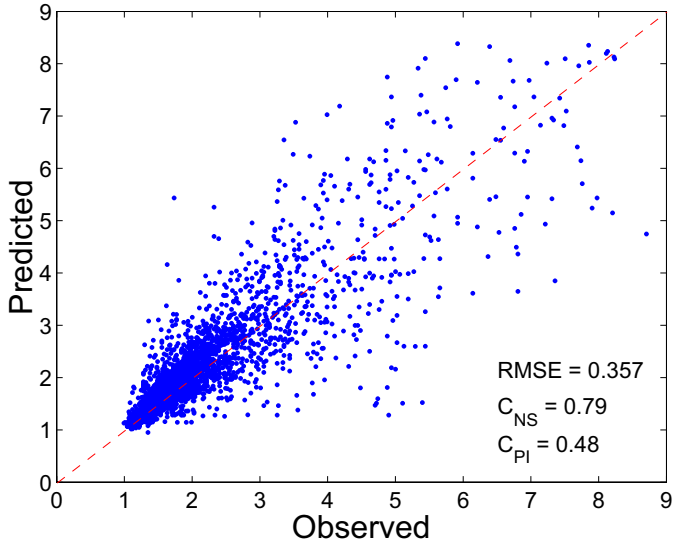


Figure 3.11: Scatter plot of predicted versus observed hourly discharges (m^3/s) for a six-hour-ahead forecast based on historical rainfall and discharge values.

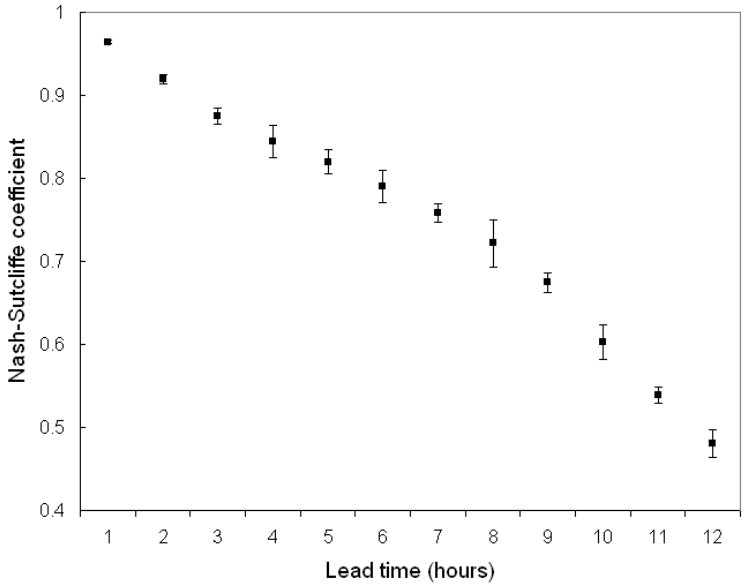


Figure 3.12: ANN performance in terms of the Nash–Sutcliffe coefficient C_{NS} (with 95% confidence bounds) for hourly multi-step-ahead predictions.

performance measures, this is indeed true for this case. As a consequence, ANN models undervalue the information contained in other input signals.

The prediction lag effect is especially significant in forecasts with small lead times, but it is also noticeable in more practically relevant forecasts with longer lead times. However, the longer the lead time L becomes, the lower the correlation between Q at t and Q at $t+L$ will be. As a result, the ANN model will give more weight to the rainfall information, which causes the prediction lags to decrease. Naturally, the overall performance in terms of squared errors also decreases with longer lead times (see Fig. 3.12). All this can be explored in more detail in Fig. 3.14, where forecast results for various lead times are evaluated in terms of C_{NS} (shown on the ordinate), and for various shifts in time of the forecasted versus the observed time series (shown on the abscissa). The ANN models that were used for these simulations are the same as in the previous simulations. The C_{NS} at zero shift corresponds to the actual performance of the models. The predicted time series is subsequently shifted in time against the observed time series, after which C_{NS} is recalculated. The time shift at which the C_{NS} coefficient is maximized, is an expression for the mean lag in the model forecast. This is done for a number of different lead times (the different lines). The idea for this method of timing evaluation is taken from Conway *et al.* [1998]. What Fig. 3.14 shows is that the prediction lag increases with the lead forecast time (i.e., the peaks are further to the left for longer lead times), but not proportionally. Another thing that can be clearly observed is the dramatic decrease in C_{NS} for longer lead times, which can be read from the vertical line at a time shift of 0. The above proves that the training on MSE or C_{NS} can be inadequate and that there is much to be gained

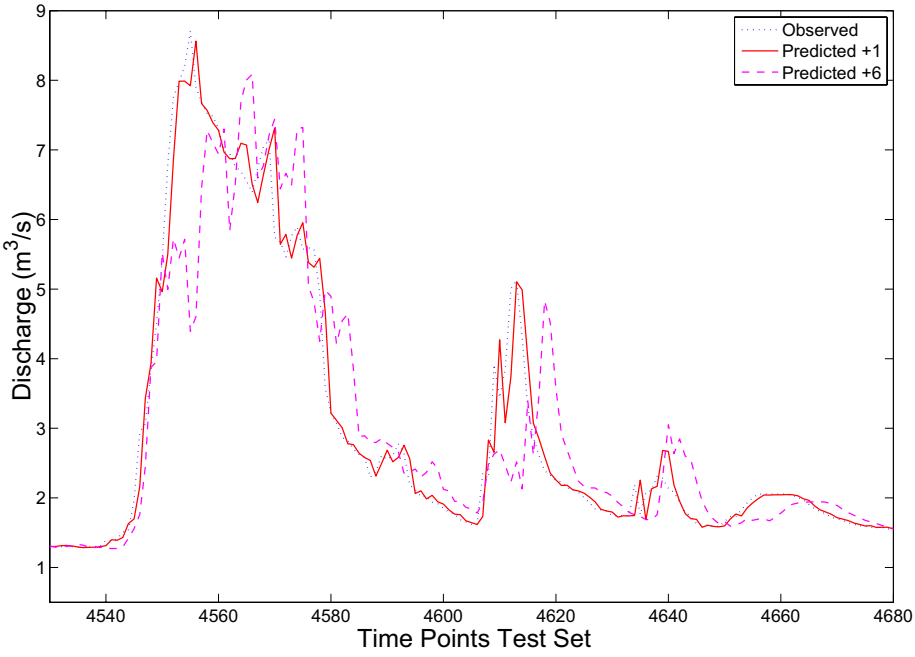


Figure 3.13: Observed and predicted hourly time series for a one-hour-ahead and a six-hour-ahead forecast (detail; July 8 to July 13, 1993).

by correcting ANN models for timing errors.

The issue of lagged predictions in ANN model forecasts, and the relation with the introduction of an autoregressive component by using previous discharge values, has been rarely addressed in literature. Only a small number of researchers have explicitly discussed timing errors (e.g., Minns [1998]) or attempted to resolve the issue (e.g., Varoonchotikul [2003]; Abrahart *et al.* [2007]). Unfortunately, no adequate method for satisfactorily dealing with prediction lags has yet been developed. Nevertheless, the problem is wide-spread, as proven by various research results that indicate that lags indeed occurred in the ANN model forecasts (e.g., Campolo *et al.* [1999]; Dawson and Wilby [1999]; Zealand *et al.* [1999]; Thirumalaiah and Deo [2000]; Gaume and Gosset [2003]; A. Jain and Srinivasulu [2004]).

The one-day-ahead forecast of the previously discussed daily-data models outperforms the forecasts of the hourly-data models with a lead time of six hours and more (both in terms of timing and C_{NS}, cf. Figs. 3.8 and 3.11). The reason for this difference in performance is that the cross-correlation between the daily rainfall and discharge series is higher than that of the hourly series, while the autocorrelation of the daily discharge series is lower than that of the hourly series (shown in Figs. 3.5 and 3.6). As a result, the information content of the daily input data is more evenly spread over the various input signals and the autoregressive component of the ANN R–R model does not become as

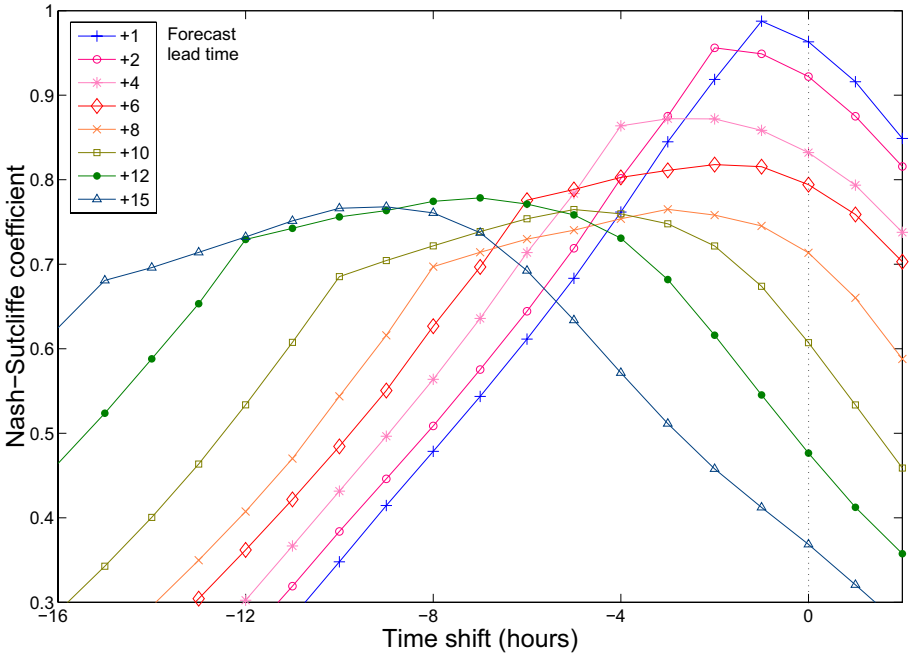


Figure 3.14: ANN multi-step-ahead forecast performances in terms of the Nash–Sutcliffe coefficient for various shifts in time of the forecasted versus the observed time series.

dominant that the forecasts show a consequent lag in time. It is important to realize that the significance of the prediction lag effect depends on the distribution of the information content of the various input time series (which is often related to the time resolution of the hydrological series). Also, the severity of timing errors also depends on the requirements of the forecasts: depending on the type of high flows that are common for a catchment, one can prefer to have a better overall approximation of the flows (e.g., in case of prolonged high flows), instead of more accurate timing (e.g., in case of flash floods).

Two sources of the prediction lag effect can be identified, each of which may be able to suggest possible solutions. Firstly, there is the matter of ANN model input. If previous discharge values are used for hydrological state representation of the system, pronounced negative effects may be introduced in the form of prediction lags. Secondly, there is the difficulty of evaluating ANN model performance, especially during the training phase. The squared-error-based performance measure that was used for model training and evaluation is clearly not always strict enough to result in a satisfactory R–R model, since it may undervalue correct timing of the forecast. Both topics are addressed in the following two sections respectively.

3.4.2 On Hydrological State Representation

The hydrological state of a river basin prior to a rainfall event is important in governing the processes by which a catchment responds to this rainfall and the proportion of the input volume that appears in the stream as part of the hydrograph [Beven, 2001b]. The majority of studies on ANNs in R–R modeling have used input signals that are merely indirectly related to these hydrological conditions. For example, previous values of discharge or water levels can be considered indirect indicators of the hydrological state of a catchment and are therefore often used as model inputs (e.g., Hsu *et al.* [1995]; Minns and Hall [1996]; Campolo *et al.* [1999]). This study proves that this may not be a good solution, because the autoregressive model component that is thus introduced can become too dominant, resulting in lagged model forecasts. Another possible source of information for the hydrological state is the (weighted) cumulative rainfall over a preceding period of time (e.g., Shamseldin [1997]; Rajurkar *et al.* [2004]). Air-temperature or (potential) evaporation time series are also often used in combination with rainfall time series (e.g., Zealand *et al.* [1999]; Tokar and Markus [2000]). These evaporation and temperature data can be considered to account for losses in the water balance of a catchment, thereby adding to the information on the hydrological state. More direct indicators of the hydrological state are variables related to soil moisture and groundwater levels. Recent studies by Gautam *et al.* [2000] and Anctil *et al.* [2004] have shown that time series of soil moisture measurements and estimations can be successfully used as ANN model input. de Vos [2003] and de Vos *et al.* [2004] have proven the value of groundwater level time series as ANN inputs.

Three alternatives for hydrological state representation are tested and compared in terms of both squared error and timing in this work. Firstly, a time series of the non-decaying moving average of the discharge (Q_{ma}) was used as ANN input. A moving average time series of the discharge can also be considered to represent the hydrological state and has the advantage over using discharge time series that the correlation with the ANN output is lower. The near absence of lags in the daily-data model forecasts and the decrease of the prediction lag effect with increased lead times (see Fig. 3.14) suggest that this approach would improve timing accuracy. Based on trial-and-error runs using this variable as ANN input for predicting discharge, a memory length of 192 hours (8 days) for the moving average of the discharge was used. Secondly, time series of the non-decaying moving average of the rainfall (P_{ma}) were constructed using a memory length of 480 hours. Lastly, the simple soil moisture reservoir component of the GR4J lumped conceptual rainfall-runoff model [Edijatno *et al.*, 1999; Perrin *et al.*, 2003] was used to produce a time series of estimated soil moisture (S). Note that this synthetic time series was generated prior to any ANN modeling and was subsequently used as ANN input as substitute for measurements related to soil moisture. The GR4J model component for soil moisture comprises a single reservoir with either net outflow in the case where the potential evaporation (E_P) exceeds

the rainfall intensity (P):

$$S^* = S_{t-1} - \frac{S_{t-1}\,(2A - S_{t-1})\tanh\left(\frac{E_{P,t}-P_t}{A}\right)}{A + (A - S_{t-1})\tanh\left(\frac{E_{P,t}-P_t}{A}\right)} \qquad \text{if} \qquad (P_t \leq E_{P,t}) \tag{3.8}$$

or net inflow in all other cases:

$$S^* = S_{t-1} + \frac{\left(A^2 - S_{t-1}^2\right)\tanh\left(\frac{P_t-E_{P,t}}{A}\right)}{A + S_{t-1}\tanh\left(\frac{P_t-E_{P,t}}{A}\right)} \qquad \text{if} \qquad (P_t > E_{P,t}) \tag{3.9}$$

where S^* can never exceed A. Finally, the percolation from the storage reservoir is taken into account using the following formula:

$$S_t = S^* \cdot \left[1 + \left(\frac{4S^*}{9A}\right)^4\right]^{-\frac{1}{4}} \tag{3.10}$$

The hourly rainfall time series and temporally downscaled potential evaporation time series served as input to the GR4J soil moisture model component. Downscaling of evaporation was simply done by taking 24 hourly values equal to the daily value. The filtering effect of the soil moisture reservoir made the inclusion of further details in the downscaling procedure, such as sinusoidal shapes for daily evaporation cycles, unnecessary. The only parameter that needed to be defined is the reservoir's maximum capacity A. Of the several values that were tested, a maximum capacity of 400 mm produced the best results. The best initial value for the storage in the reservoir was found to be 180 mm. The result is presented in Figure 3.15. Anctil *et al.* [2004] have also used the GR4J model component to create soil moisture time series, which too were subsequently used as ANN input. Reference is made to their interesting paper, which gives a more extensive and in-depth presentation on the topic of combining soil moisture modeling with ANN R–R modeling.

Tables 3.2 and 3.3 show that the simulations with P_{ma} and the S time series are not affected by any prediction lags. The performance as indicated by the C_{NS} and C_{PI}, however, is mediocre and only slightly better than using only the P time series as ANN input. Using the Q_{ma} time series results in decreased (but still noticeable) prediction lags compared to the simulations with Q, but the C_{NS} and C_{PI} also decrease. Similar C_{NS} and C_{PI} results are produced by a combination of P_{ma}, Q_{ma} and S, but the prediction lag effect is almost eliminated. It is interesting to note that the test results show that any combination of these variables with Q still results in prediction lags, suggesting that the autoregressive component again dominates as a result of using Q as ANN input. In the case of six-hour-ahead forecasts, however, the average prediction lag decreases from -2 to -1 due to the additional information in the P_{ma}, Q_{ma} and S model inputs. This

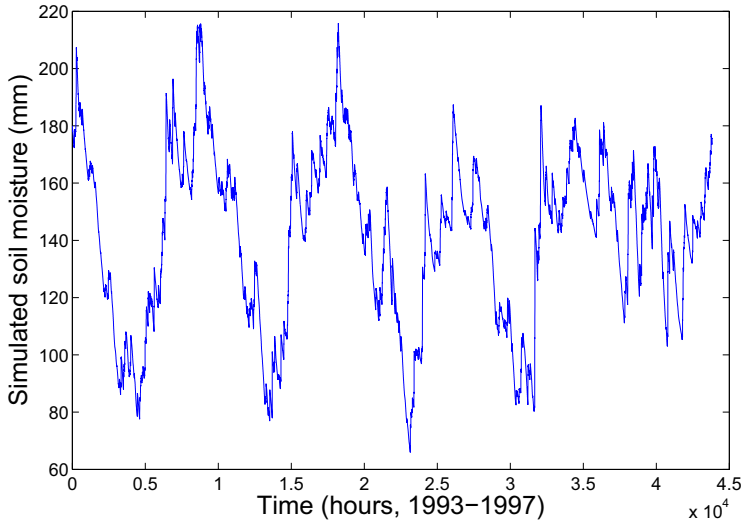


Figure 3.15: Time series of simulated soil moisture using the GR4J soil moisture component.

Table 3.2: ANN model performance for one-hour-ahead forecast using various methods of hydrological state representation (results over 10 training trials).

Input	Time window	Config.	$\overline{C_{NS}}$	$\sigma_{C_{NS}}$	$\overline{C_{PI}}$	$\sigma_{C_{PI}}$	Avg. lag
P	−5 to −19	15–4–1	0.513	0.047	−10.676	1.134	0.1
P	−5 to −19	18–5–1	0.963	0.001	0.121	0.020	−1.0
Q	0 to −2						
P	−5 to −19	18–5–1	0.803	0.020	−3.557	0.494	−1.0
Q_{ma}	0 to −2						
P	−5 to −19	18–5–1	0.479	0.057	−11.403	1.398	0.0
P_{ma}	0 to −2						
P	−5 to −19	18–5–1	0.560	0.022	−9.540	0.535	0.0
S	0 to −2						
P	−5 to −19	24–5–1	0.656	0.044	−7.238	1.054	−0.1
Q_{ma}	0 to −2						
P_{ma}	0 to −2						
S	0 to −2						
P	−5 to −19	27–5–1	0.964	0.002	0.133	0.035	−1.0
Q	0 to −2						
Q_{ma}	0 to −2						
P_{ma}	0 to −2						
S	0 to −2						

Table 3.3: ANN model performance for six-hour-ahead forecast using various methods of hydrological state representation (results over 10 training trials).

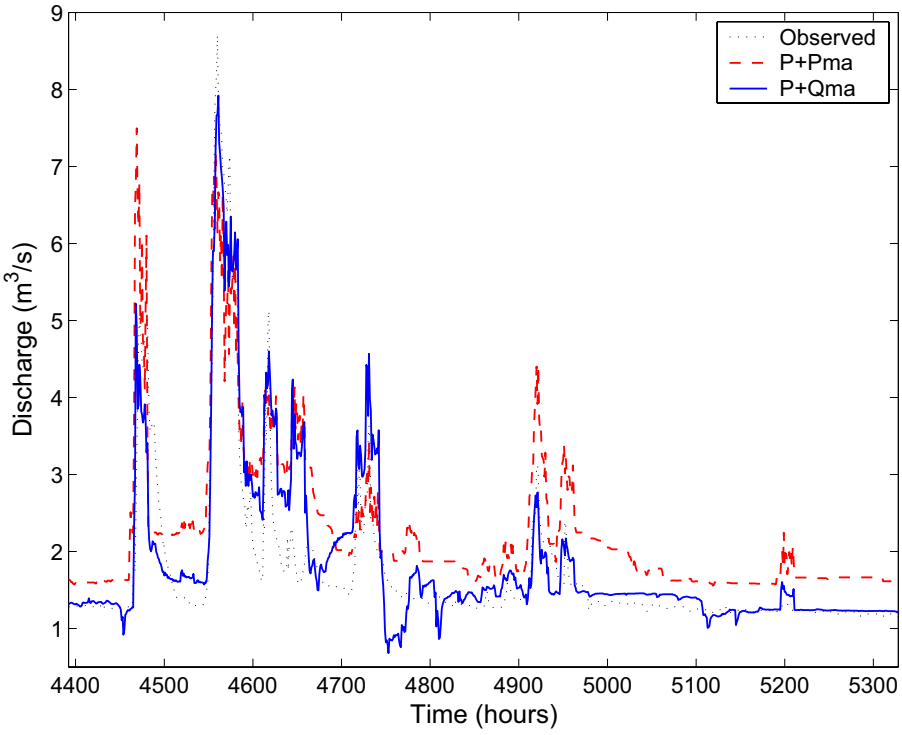
Input	Time window	Config.	$\overline{C_{NS}}$	$\sigma_{C_{NS}}$	$\overline{C_{PI}}$	$\sigma_{C_{PI}}$	Avg. lag
P	0 to −14	15–4–1	0.491	0.032	−0.258	0.079	0.0
P	0 to −14	18–5–1	0.791	0.006	0.482	0.015	−2.0
Q	0 to −2						
P	0 to −14	18–5–1	0.682	0.012	0.213	0.029	−0.8
Q_{ma}	0 to −2						
P	0 to −14	18–5–1	0.521	0.061	−0.185	0.150	0.0
P_{ma}	0 to −2						
P	0 to −14	18–5–1	0.558	0.054	−0.092	0.134	0.0
S	0 to −2						
P	0 to −14	24–5–1	0.688	0.016	0.229	0.039	−0.1
Q_{ma}	0 to −2						
P_{ma}	0 to −2						
S	0 to −2						
P	0 to −14	27–5–1	0.806	0.014	0.518	0.035	−1.0
Q	0 to −2						
Q_{ma}	0 to −2						
P_{ma}	0 to −2						
S	0 to −2						

proves that even strongly dominant autoregressive model components can be suppressed by using additional input signals, resulting in better forecast timing.

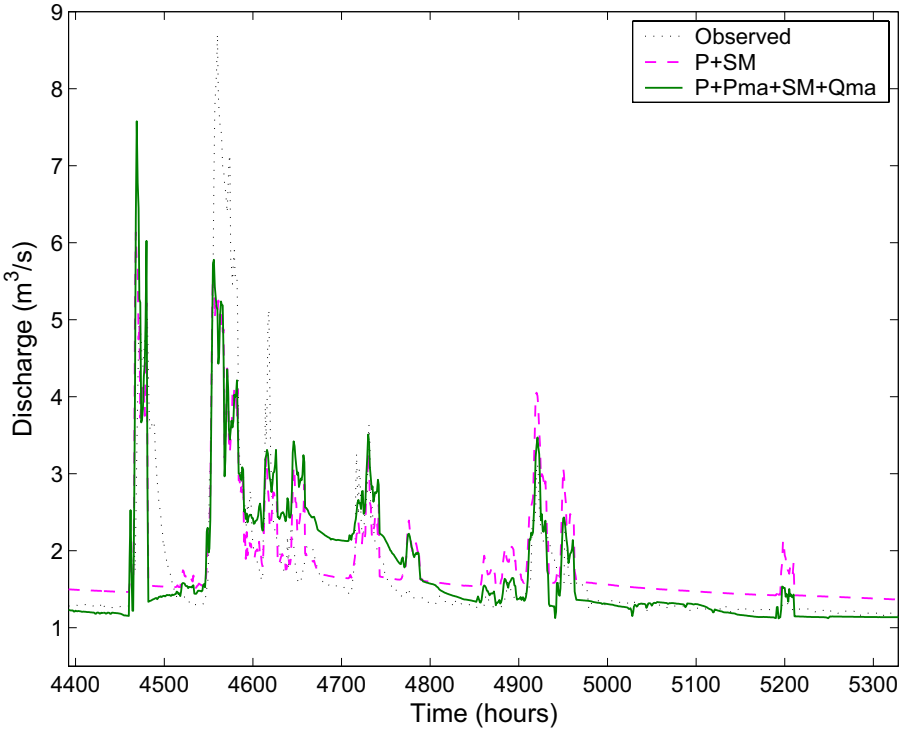
Figures 3.16 and 3.17 present details of the forecasted time series using the various hydrological state representations. The simulations with P_{ma} in Figs. 3.16(a) and 3.17(a) show a consistent overestimation of low flows and an inaccurate reproduction of the shape of the recession curves. Moreover, most peak flows are underestimated, especially in the six-hour-ahead forecast. The models with S (Figs. 3.16(b) and 3.17(b)) underestimate high peak flows, but reproduce low flows and recession curves reasonably well (although with a slight overestimation). There are abrupt changes in the slope of the recession curve, however, where a more gradual decrease of the discharge would be more accurate. This is probably a result of using the simple GR4J model for creating the S time series, and other soil moisture models or soil moisture measurements might produce better results. The ANNs that used Q_{ma} as input (Figs. 3.16(a) and 3.17(a)) show good overall performance but are subject to some inaccuracy due to fluctuations that occurred in periods of low flows. They are best at simulating peak flows, even though more than half of the peaks were still underestimated significantly (by 10% or more). Neither of the three alternatives can be considered very adequate as sole representation of hydrological state. However, the simulations with all three alternatives for hydrological state representation (i.e., P_{ma}, Q_{ma}, S) show that the ANN model attempts to combine the best of each alternative (see Figs. 3.16(b) and 3.17(b)). This can be concluded from the reasonably good overall performance (mainly resulting from the Q_{ma} input) and the correctly timed forecasts (mainly resulting from the P_{ma} and S inputs). A visual inspection shows that for the one-hour-ahead forecast, the information from all input signals is approximately equally weighted, and the six-hour-ahead forecast is slightly dominated by the information contained in Q_{ma}. Figure 3.18 shows scatter plots of the one-hour-ahead and the six-hour-ahead forecasts for this model type.

Note that in neither of the above simulations extreme peak flows are well approximated. One of the reasons for this is that ANN models have difficulties dealing with the extremely nonlinear catchment response in the case of wet catchment conditions in combination with rainfall events. Another reason is that these ANN models attempt to simulate the complete range of the hydrograph and therefore may undervalue the high peak flow errors, since these flows occur only incidentally. Moreover, there are only a few examples of extreme peak flows in the training data, and hence the model has only little information on these types of events to which it can adapt.

Finding better ways of representing hydrological state is an important step toward better ANN modeling of R–R processes. The various ANN input signals that serve as a way of state representation can complement each other in terms of information content, but they are also likely to have some information overlap. The ability to exploit the total

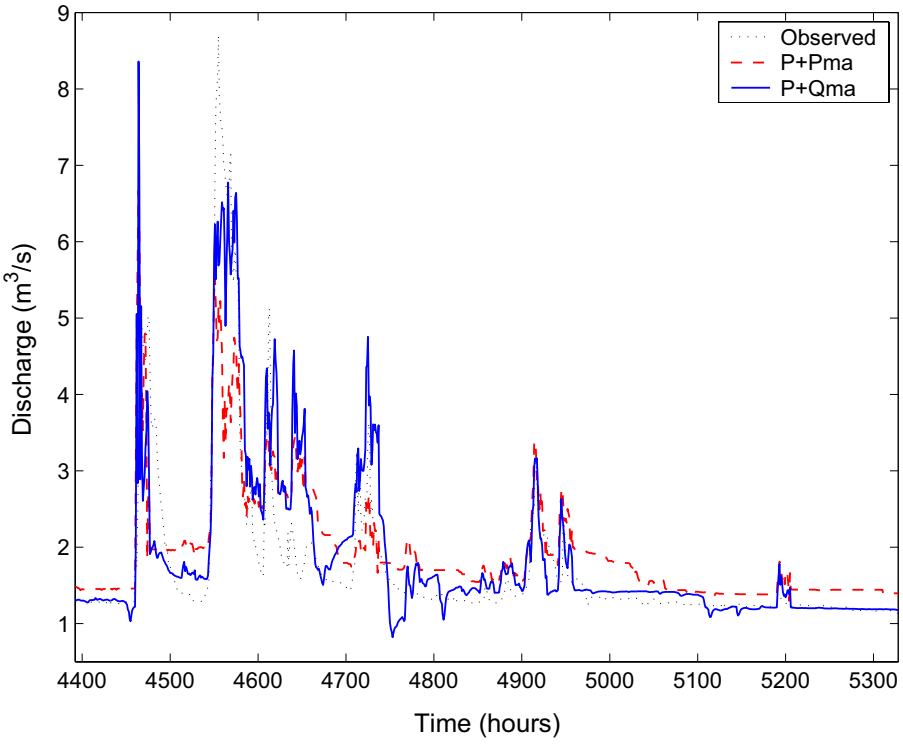


(a)

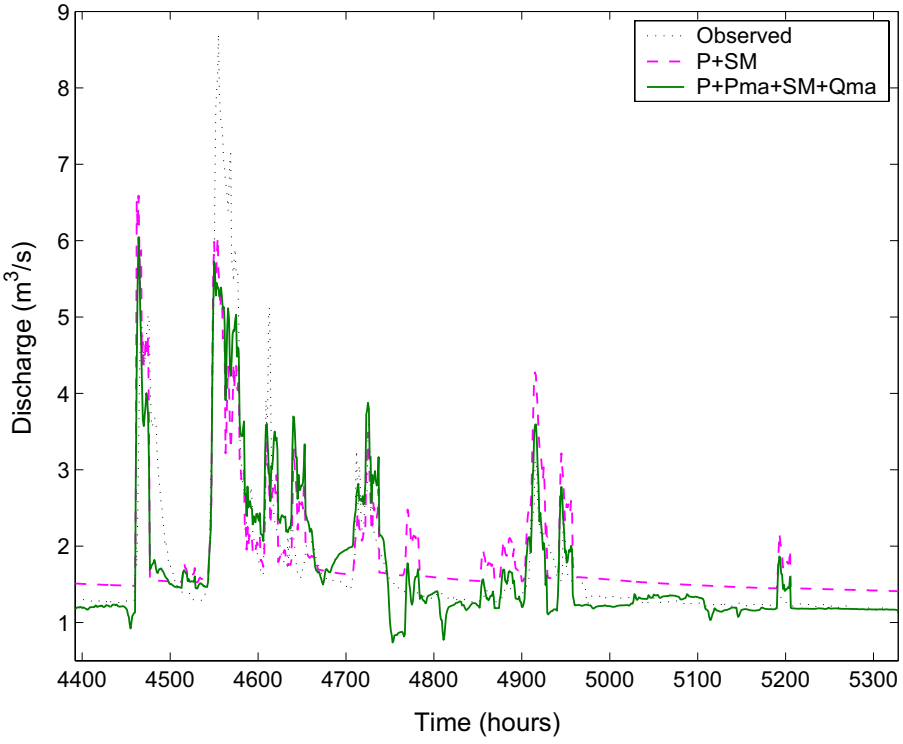


(b)

Figure 3.16: ANN model results for one-hour-ahead forecasted discharge time series (m^3/s) using various methods of hydrological state representation (detail; July 3 to August 11, 1993).

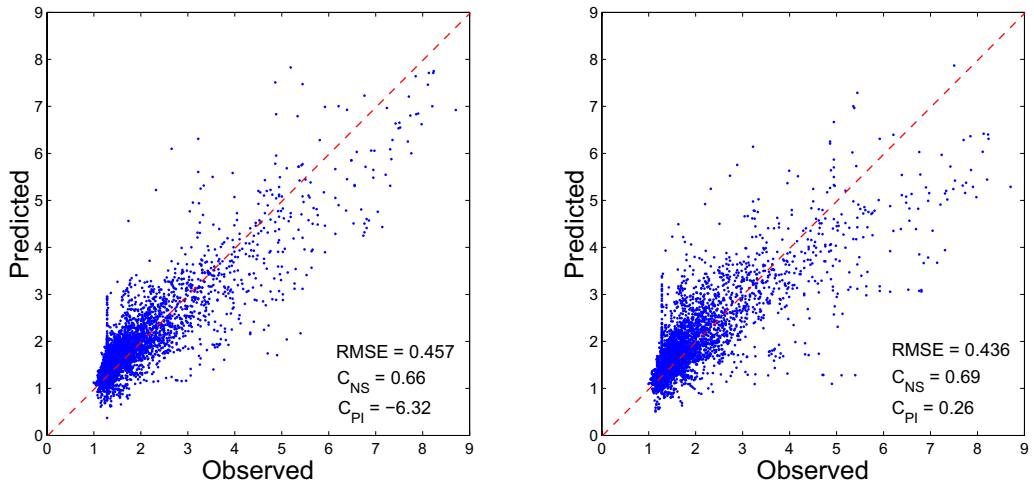


(a)



(b)

Figure 3.17: ANN model results for six-hour-ahead forecasted discharge time series (m^3/s) using various methods of hydrological state representation (detail; July 3 to August 11, 1993).



(a) (b)

Figure 3.18: Scatter plots of predicted versus observed hourly discharges (m^3/s) for (a) a one-hour-ahead and (b) a six-hour-ahead forecast based on P, P_{ma}, S, and Q_{ma} inputs.

information content depends strongly on the training algorithm and the performance measure that this algorithm is trying to optimize. The following section will discuss the choice of performance measures in ANN training for R–R modeling.

3.4.3 Performance Measures for ANN Training

In order to prevent prediction lags, an aggregated objective function was tested that punishes the ANN model for having a timing error. The timing error used is defined as the time shift of the entire forecast time series for which the RMSE is at a maximum (i.e., the peak in Figure 3.14), and is therefore a measure of the overall timing error of the model. The time shifts over which this check is performed varied from -20 to $+20$ time steps.

Training was performed using aggregated objective functions, which consist of combinations of the products of the RMSE, the MSLE and a timing error factor (TEf). The TEf is 500 if the timing error is non-zero and 1 if the timing error is zero. The value of 500 was selected by trial and error from a set of arbitrarily chosen numbers. This way the model is penalized for having a timing error other than zero. The idea for this method is taken from Conway *et al.* [1998] who used a Genetic Algorithm to train ANN models for predicting solar activity.

The ANN performance results in terms of mean and standard deviation over the best 40 out of 50 training trials for lead times of one and six hours are presented in Tables 3.4 and 3.6. The 10 worst-performing trials were deleted because they are outliers that are not representative for the ANN model behavior. The results for the models trained using the

Table 3.4: Evaluation performance of one-hour-ahead forecasts of ANN models that are trained with various objective functions, including a timing correction factor. Presented are the mean and standard deviation over 40 simulations.

Obj. Function	C_{NS}	C_{PI}	TE	MSLE
C_{NS}	0.96 ± 0.00	0.13 ± 0.04	-1.00 ± 0.00	24.6 ± 2.3
MSLE	0.96 ± 0.00	0.13 ± 0.02	-1.00 ± 0.00	24.4 ± 1.1
$C_{NS} \cdot$ TEf	0.52 ± 0.48	-10.5 ± 11.5	-0.89 ± 0.39	750 ± 800
MSLE $\cdot$ TEf	0.62 ± 0.37	-8.2 ± 9.0	-0.90 ± 0.50	586 ± 816
$C_{NS} \cdot$ MSLE $\cdot$ TEf	0.89 ± 0.11	1.55 ± 2.56	-1.00 ± 0.00	115 ± 147

Table 3.5: Evaluation performance of one-hour-ahead forecasts of ANN models that are trained with various objective functions, including a timing correction factor. Presented are only the best models.

Obj. Function	C_{NS}	C_{PI}	TE	MSLE
C_{NS}	0.97	0.23	-1.00	21.6
MSLE	0.97	0.17	-1.00	22.1
$C_{NS} \cdot$ TEf	0.96	0.12	-1.00	22.8
MSLE $\cdot$ TEf	0.96	0.16	-1.00	23.8
$C_{NS} \cdot$ MSLE $\cdot$ TEf	0.97	0.20	-1.00	21.7

Table 3.6: Evaluation performance of six-hour-ahead forecasts of ANN models that are trained with various objective functions, including a timing correction factor. Presented are the mean and standard deviation over 40 simulations.

Obj. Function	C_{NS}	C_{PI}	TE	MSLE
C_{NS}	0.80 ± 0.01	0.51 ± 0.02	-1.05 ± 0.22	130 ± 9
MSLE	0.80 ± 0.01	0.49 ± 0.02	-1.15 ± 0.36	138 ± 9
$C_{NS} \cdot$ TEf	0.18 ± 0.57	1.04 ± 1.43	-0.93 ± 1.87	1410 ± 2290
MSLE $\cdot$ TEf	0.31 ± 0.42	0.71 ± 1.05	-0.93 ± 1.87	875 ± 656
$C_{NS} \cdot$ MSLE $\cdot$ TEf	0.46 ± 0.38	0.35 ± 0.93	-0.61 ± 1.16	632 ± 513

Table 3.7: Evaluation performance of six-hour-ahead forecasts of ANN models that are trained with various objective functions, including a timing correction factor. Presented are only the best models.

Obj. Function	C_{NS}	C_{PI}	TE	MSLE
C_{NS}	0.82	0.55	−1.00	123
MSLE	0.82	0.56	−1.00	115
$C_{NS} \cdot$ TEf	0.78	0.46	0.00	150
MSLE $\cdot$ TEf	0.78	0.45	0.00	158
$C_{NS} \cdot$ MSLE $\cdot$ TEf	0.79	0.47	0.00	141

timing error factor in the objective function show an improvement in timing only at the cost of a degradation of other performance measures. The single best results out of these trials according to expert judgment as presented in Tables 3.5 and 3.7, however, show some promising results. Apparently, some solutions were found for a lead time of 6 hours in which the ANNs are capable of making correctly timed forecasts, while maintaining reasonably good performance in terms of other statistics. This proves that for six-step-ahead forecast models, in which the influence of the last recorded observation is less than in the one-step-ahead models, it is possible to find good optima that have correct forecast timing in combination with good overall fit. The one-step-ahead forecasts seemed not to be affected by the measures to prevent timing errors.

Unfortunately, the LM training algorithm has more difficulty in finding optima when implementing timing in the objective function. This is probably due to the introduction of the multiplication factor TEf, which makes the gradients of the objective functions extremely irregular. The use of optimization algorithms that do not rely on gradients such as the Genetic Algorithm, as used by Conway *et al.* [1998]) might alleviate this problem. Further research on timing errors in ANN R–R modeling and on possible solutions was conducted by Abrahart *et al.* [2007].

3.5 Summary and Discussion

The purpose of this chapter was to find whether multi-layer, feedforward ANNs can be effectively used as R–R models, and to investigate the role of hydrological state representation in ANN R–R modeling. The results of the one-day-ahead forecasts using daily data were promising and in accordance with the consensus that (at least in some cases) ANNs are good alternatives for traditional R–R modeling approaches. However, the simulations with hourly data were afflicted by lags in the ANN model forecasts. Previously observed values of discharge are often used as ANN model inputs, since they are considered indicators of the hydrological state. Such data, however, introduce an autoregressive model

component in the ANN model. The results show that high autocorrelation of the discharge time series may result in an uneven spread of the information content in network input. This leads to the autoregressive model component becoming too dominant and the ANN model producing a forecast that is very similar to the last known discharge, effectively causing timing errors in the predictions. The prediction lag effect is especially significant for short lead times, but also forecasts with longer lead times were affected by it. This issue was discussed from two points of view: (1) hydrological state representation and (2) model performance measures for ANN training. Firstly, instead of representing the hydrological state using previous discharge, a number of alternatives was tested. The best results, in terms of timing and overall fit, were obtained using a combination of hydrological state representations: a moving average over the previous discharge, a moving average over the previous rainfall, and the output of the simple GR4J soil moisture model. The usefulness of the latter proves that complementary conceptual models can be valuable additions to ANN model approaches. Secondly, it is concluded that not all differences between modeled and observed hydrograph characteristics (e.g., timing, peak values, volume) can be adequately expressed by a single performance measure such as the MSE. The use of a timing error statistic during ANN training as a method of increasing timing accuracy of ANN rainfall-runoff model forecasts showed to be only partly effective since the performance according to other performance measures is decreasing. There seems to be a trade-off between the objectives of correct timing and good overall fit. However, some weight sets were found that indicate the possibility of finding an acceptable compromise between the two. The results from this chapter suggest ANN models will benefit from using more than one performance measure in their training.

Chapter 4

Multi-Criteria Training of Artificial Neural Network Rainfall–Runoff Models

Modified from:

de Vos, N. J., Rientjes, T. H. M., 2008. Multi-objective training of artificial neural networks for rainfall–runoff modeling. *Wat. Resour. Res.* **44**, W08434.

Abstract This chapter presents results on the application of various optimization algorithms for the training of artificial neural network rainfall-runoff models. Multi-layered feedforward networks for forecasting discharge from two meso-scale catchments in different climatic regions have been developed for this purpose. The performances of the multi-criteria algorithms MOSCEM–UA and NSGA–II have been compared to the single-criterion Levenberg–Marquardt and Genetic Algorithm for training of these models. Performance has been evaluated by means of a number of commonly applied objective functions and also by investigating the internal weights of the networks. Additionally, the effectiveness of a new objective function called Mean Squared Derivative Error, which penalizes models for timing errors and noisy signals, has been explored. The results show that the multi-criteria algorithms give competitive results compared to the single-criterion ones. Performance measures and posterior weight distributions of the various algorithms suggest that multi-criteria algorithms are more consistent in finding good optima than single-criterion algorithms. However, results also show that it is difficult to conclude if any of the algorithms is superior in terms of accuracy, consistency and reliability. Besides the training algorithm, network performance is also shown to be sensitive to the choice of objective function(s), and including more than one objective function proves to be helpful in constraining the neural network training.

4.1 Introduction

R–R models are often calibrated using a single objective function that aggregates the difference between an observed and a simulated time series such as river discharge. Various

researchers, however, have argued that the model calibration problem is inherently multi-criteria (MC) and that the single-criterion (SC) paradigm signifies a loss of information with respect to the original hydrological signal (see Gupta *et al.* [1998]). SC calibration therefore can be considered inappropriate if the dimension of the parameter space greatly exceeds the single dimension of the objective function [Gupta *et al.*, 2008]. Especially when using automatic calibration algorithms, there is increased risk of finding parameter values that are physically unrealistic and that compensate for faulty values of other parameters, but also measurement errors and model structural errors [Sorooshian and Gupta, 1983b; Boyle *et al.*, 2000; Madsen, 2000]. MC calibration, on the other hand, can use multiple model outputs such as river discharge, stream chemistry or storage variables, or a single output using multiple objectives that reflect specific characteristics.

The use of multiple objectives has generally focused on so-called preference-based methods, in which weights are assigned to objective functions and the MC problem is simplified as a SC one (e.g. [Madsen, 2000; Seibert, 2000; Cheng *et al.*, 2002; Seibert and McDonnell, 2002; Rientjes, 2004]). Studies are also presented that use no-preference MC algorithms for conceptual R–R model calibration (e.g., [Gupta *et al.*, 1998; Yapo *et al.*, 1998; Boyle *et al.*, 2000; Vrugt *et al.*, 2003a; Khu and Madsen, 2005; Kashif Gill *et al.*, 2006; Tang *et al.*, 2006; Fenicia *et al.*, 2007a; de Vos and Rientjes, 2007, 2008b]). These studies show that MC model calibration is effective in knowledge-based R–R modeling in the sense that information contained in the data series is used more effectively, which generally leads to improved model performance. Additionally, some insight is gained into why and under what circumstances models fail (e.g., [Gupta *et al.*, 1998; Boyle *et al.*, 2000]). Still, it is found that there is often a clear trade-off between model performance on different response modes of a catchment as a result of inadequacies in the functioning of a model [Wagener *et al.*, 2003b].

This chapter presents a study on MC calibration in data-driven ANN modeling for R–R simulation. Although model structures from ANN models differ fundamentally from those of conceptual models, training of these models is in essence similar: model parameters are optimized by minimizing residual errors that represent the mismatch between model output and observed data. While in conceptual modeling parameters often have some interpretable physical meaning, in data driven R-R modeling such interpretation is commonly missing. Recent research, however, on the ANN by for instance Wilby *et al.* [2003]; Sudheer and Jain [2004]; A. Jain and Srinivasulu [2006] show that some physically interpretable information can be found in ANN weights and hence confirm the similarities between ANN training and conceptual model calibration. It is therefore reasoned that the MC paradigm is also applicable to ANN R–R modeling and that it could lead to better extraction and utilization of information in available time series. The hypothesis is that an MC algorithm finds major optima on the response surface of multiple objective functions, making it more likely to find a stable region that allows for consistency in model

performance and thus improves model reliability.

The goals of this chapter are therefore (1) to test MC optimization algorithms for the training of ANN R–R models (2) to assess effectiveness when compared to traditional SO algorithms, (3) to test a number of combinations of objective functions for training of the ANNs and (4) to compare the *a posteriori* weight distribution of ANNs after both SC and MC training. ANN models are developed for the Leaf River basin and the Geer River basin (see Appendix A). These models are trained with the SC Levenberg–Marquardt (LM) and Genetic Algorithm (GA) algorithm and the MC NSGA–II and MOSCEM–UA algorithms.

4.2 Artificial Neural Network Model Description

4.2.1 Input

The ANN models used in this chapter are again feedforward networks with one hidden layer of neurons (see Figure 4.1 for an example). The transfer function that was used in both the hidden and the output layer is the logistic function (Equation 4.1). Because of the saturation and the output range of this function, all input data were linearly scaled between -1 and 1 and the output data between 0 and 1.

$$f(x) = \frac{1}{1 + \exp(-x)} \tag{4.1}$$

4.2.2 Training

Just as in Chapter 3, the ANN models are trained based on a so-called supervised training procedure which allows the network to simulate the hydrological system by examining input-output examples from it. Work by Samani *et al.* [2007] and also the previous chapter show that the popular steepest-descent backpropagation algorithm is sometimes easily outperformed by second-order gradient algorithms and a wider consensus has been reached that such algorithms are therefore preferable over first-order methods.

Any gradient-based algorithm, however, still commonly suffers from the issue that it is essentially a local search method. It therefore carries a significant risk of getting stuck in local optima. Research by Duan *et al.* [1992], Goldberg [2000] and [Deb, 2001] shows the effectiveness of global, evolutionary-based algorithms in parameter estimation. The Genetic Algorithm (GA) is the most popular evolutionary algorithm and has been successfully applied to ANN training by, for example Rooij *et al.* [1996] and Sexton *et al.* [1998]. It has also been used for so-called neuro-evolution, in which the dual problem of parameter estimation and model structural optimization is solved simultaneously (e.g.

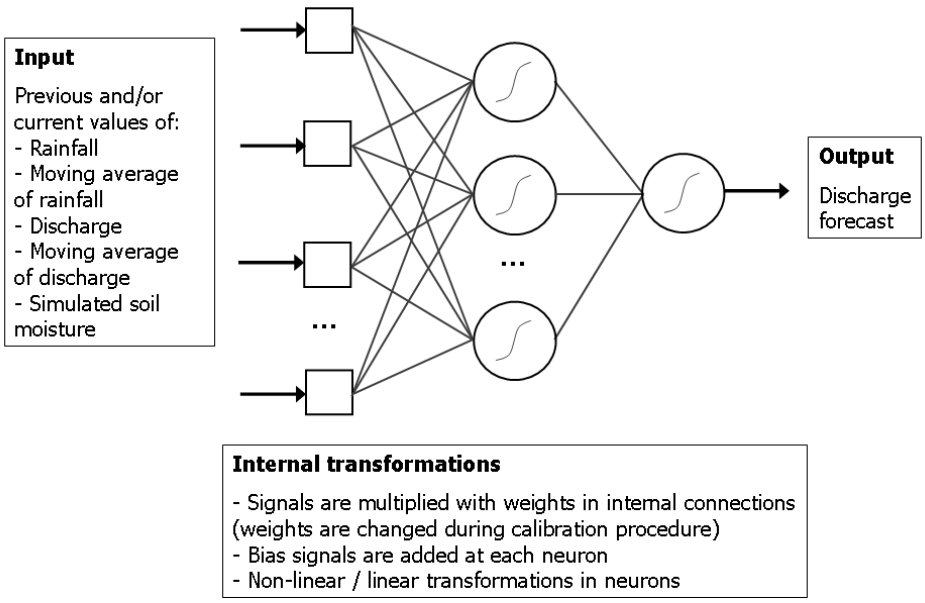


Figure 4.1: Example of the type of feedforward ANNs used in this study. This is an ANN with one hidden layer and one output neuron.

Dawson *et al.* [2006]). As opposed to gradient-based algorithms, it is shown that optimization on evolutionary principles generally performs better in terms of accuracy and consistency, although often at the expense of extra computational efforts. A description on principles of a GA is ignored here and reference is made to the textbook of Goldberg [2000].

SC algorithms that are tested in this research are the LM algorithm and GA. The standard implementation of LM in the MATLAB Neural Network Toolbox was used, with no memory reduction settings. The GA as implemented in the Genetic Algorithm Optimization Toolbox [Houck *et al.*, 1995] was used, with tournament selection, arithmetic crossover and non-uniform mutation. The two evolutionary-based MC algorithms that are used are NSGA–II [Deb *et al.*, 2002] and MOSCEM–UA [Vrugt *et al.*, 2003a], and they are discussed in more detail in Section 4.3. All evolutionary algorithms optimize weights in the range between -8 and 8, which was considered sufficiently large to find reasonable solutions. The LM algorithm was not bounded in its search range.

Randomness is introduced in the ANNs initialization, in which normally distributed random values for the network weights are generated. Additionally, the evolutionary algorithms occasionally use random operations in their procedure. The outcome of the resulting randomness may be that different objective function optima are found for each optimization run. This variability in parameter estimates can be interpreted as a measure

of uncertainty of the combination of ANN model and training algorithm. Since randomness may have a pronounced effect on model performance in this research, all algorithms are run over ensembles. For the SC algorithms the weights are independently re-initialized and trained 20 times, while for the MC algorithms this ensemble size is set to 10. The smaller number was chosen because a single MC algorithm run generally already produces a significant number of solutions.

4.2.3 Evaluation

In this research, six numerical performance measures are considered for ANN model evaluation of which three are thought to give an expression of overall fit: the MRE, C_{NS} and C_{PI}. The other three (M4E, MSLE and MSDE) are meant to evaluate specific characteristics of a hydrograph. The following two paragraphs discuss the measures in each of the two groups. Mathematical descriptions of the performance measures that were not already presented in Chapter 3 are shown below. In these equations, K is the total number of data elements, Q_k and $\hat{Q}_k$ are the observed and the simulated discharges at the kth time interval respectively.

$$MRE = \frac{1}{K}\sum_{k=1}^{K}\frac{\left|\hat{Q}_k - Q_k\right|}{Q_k} \tag{4.2}$$

$$M4E = \frac{1}{K}\sum_{k=1}^{K}\left(\hat{Q}_k - Q_k\right)^4 \tag{4.3}$$

$$MSDE = \frac{1}{K}\sum_{k=1}^{K}\left(\left(\hat{Q}_k - \hat{Q}_{k-1}\right) - \left(Q_k - Q_{k-1}\right)\right)^2 \tag{4.4}$$

The Mean Relative Error (MRE) is a relative indicator of absolute model errors. The well-known Nash–Sutcliffe coefficient of efficiency (C_{NS}) [Nash and Sutcliffe, 1970] and the Persistence Index (C_{PI}) [Kitanidis and Bras, 1980] scale the mean squared error and are therefore more indicative of performance on high flows. The C_{PI} is especially useful when previous discharge values are used as input to an ANN model since it evaluates models in comparison to a persistence model, which is a model that presents the last observation as a prediction (see [Anctil *et al.*, 2004; de Vos and Rientjes, 2005]). The C_{NS} and C_{PI} are not used here as objective functions during training but only serve as a performance indicator after training on other objective functions.

The mean fourth-power error (M4E) is considered an indicator of goodness-of-fit to peak flows, since large residuals are given a lot of importance. The mean squared logarithmic error (MSLE), which is based on the logarithmic function by Hogue *et al.* [2000] (also see Fenicia *et al.* [2006]) is more suitable for low flows due to the logarithmic transformation.

In de Vos and Rientjes [2007], the Mean Squared Derivative Error (MSDE) objective function is proposed. It expresses the difference between the first-order derivatives of the simulated and the observed discharge, which is equal to the difference in residuals between two successive time steps. The MSDE serves as an indicator of the fit of the shape of the hydrograph, and it specifically penalizes for timing errors and noise [de Vos and Rientjes, 2007]. Since this objective function does not take into account absolute differences but only the shapes of the simulated and observed hydrographs, it should be used in combination with residual-based functions such as the MRE or M4E. If only the MSDE was used for model calibration, it would result in a model that approximates the shape of the hydrograph but possibly has a large shift in flow magnitude. Note that the MSDE is related to the well-known statistic that counts the number of sign changes in the sequence of residuals, used by the National Weather Service [Brazil, 1988].

4.3 Multi-Criteria Training of ANN Rainfall-Runoff Models

4.3.1 Single-Criterion versus Multi-Criteria

In a SC model calibration approach, model performance is expressed by a single objective function that reflects a subjective choice of highlighting a specific aspect of the hydrograph. This objective function is then optimized to find what is regarded as the optimal model parameters. MC methods, on the other hand, reveal a set of solutions that represent the trade-off between the objectives involved, which is often referred to as the Pareto front. This front is commonly visualized in two-dimensional Pareto plots. The benefit of this approach is that more information from the data is used in the evaluation of the model, and if a model performs well on multiple objectives it implies performance consistency and thus the model is likely to be more reliable. Additionally, having identified MO trade-off solutions, the choice of which solution is preferred has become a more objective one [Deb, 2001]. Finally, the nature of the trade-off between various objectives reveals information on the adequacy of the model structure and parameters under investigation.

The above has been investigated in conceptual R–R modeling but not as much in data-driven R–R modeling. In ANN R–R modeling many different model structures can be selected, and the structures commonly have more weights than conceptual models have parameters. Moreover, given the black box nature of ANN models, weights are commonly thought to have little direct relation to real-world properties or measurable quantities, which makes the *a priori* estimation of their reasonable ranges difficult. It is for these reasons that ANN models are prone to the drawbacks that could arise when the training procedure is simplified by using a single objective, perhaps even more so than knowledge-based hydrological models.

A literature review reveals that MC training of ANN R–R models has received little attention and that its potential is not well assessed. In other research fields, however, a small number of studies report on applications of MO algorithms in ANN model training. Examples include Albuquerque Teixeira *et al.* [2000], Abbass [2003], Jin *et al.* [2004], and Giustolisi and Simeone [2006], who all focused on simultaneous minimization of output errors and optimization of the complexity of ANN model structure. The goal of using the latter was to either find an optimal ANN architecture or to prevent overtraining of the network. This work differs from such approaches in that it uses fixed ANN model structures and that SC and MC training algorithms are tested for various combinations of objective functions.

4.3.2 Multi-Criteria Algorithm Descriptions

In the following two paragraphs the NSGA–II and MOSCEM–UA algorithms are briefly introduced. Both are based on evolutionary search procedures and are designed to solve MC optimization problems. For detailed descriptions reference is made to the original papers mentioned below and to the work of Tang *et al.* [2006] who tested and compared MC evolutionary algorithms for calibration of conceptual R–R models.

The Non-dominated Sorting Genetic Algorithm II (NSGA–II) is proposed and discussed in [Deb, 2001] and [Deb *et al.*, 2002]. It uses the following evolutionary operators to create an offspring population from the original parent population: binary tournament selection, simulated binary crossover and polynomial mutation. The new population is selected from the parent and offspring population by sorting individuals based on ranks that express their degree of non-domination. In case of equal non-domination ranks, individuals in lesser crowded regions of the Pareto space are preferred over the other individuals in order to preserve the diversity of the population. The most important settings of the NSGA–II algorithm are the population size and number of generations, and they were chosen based on both experience with the algorithm and on trial-and-error. For all simulations of ANN1 (i.e. the Leaf River basin model), NSGA–II uses 80 as population size and $1,200$ for number of iterations. For simulations with the more parsimonious ANN2 (i.e. the Geer River basin model), a population size of 60 and 800 iterations has been selected, reducing the number of function evaluations by a factor 2. The same settings are applied to the SC GA optimization to make the comparison between the algorithms a fair one. Other settings that are kept constant throughout this study are the probabilities of crossover and mutation, which are set to 0.9 and 0.05 respectively, and the crossover and mutation distribution indices, which are both set to 20. These values are found by testing some common values as suggested by [Deb, 2001].

The MOSCEM–UA is developed by Vrugt *et al.* [2003a] and is based on the Shuffled Complex Evolutionary (SCE–UA) algorithm [Duan *et al.*, 1992]. It takes a uniformly

distributed initial population of points and ranks and sorts them according to a fitness assignment concept that is based on the work of Zitzler and Thiele [1999]. From the population, a number of complexes are constructed for which parallel sequences are started. These sequences iteratively evolve the complexes based on the probabilistic covariance-annealing method of the SCEM–UA algorithm [Vrugt *et al.*, 2003b] to avoid clustering of solutions in the most compromised region among the objectives. Finally, new complexes are formed through a process of shuffling. The algorithm's most important settings, the population size and the number of complexes, were again chosen based on experience and trial-and-error. For the ANN1 simulations the MOSCEM–UA algorithm uses 20 complexes, 2,400 random samples and 100,000 draws. For ANN2 MOSCEM–UA uses 16 complexes, 1,600 samples and 60,000 draws. Other settings are the number of evolutionary steps before reshuffling (set at the number of points in each complex divided by 4) and a scaling factor that determines the acceptance of new population members during the evolution of the complexes (set at 0.5). These values are equal to the ones used by [Vrugt *et al.*, 2003a].

4.3.3 Combinations of Objective Functions

The set of objective functions that is used during MC calibration should ideally measure different aspects of the differences between observed data and model simulations, so as to extract as much useful information as possible from the data [Gupta *et al.*, 1998]. Examples from the literature of objective function combinations that are based on a distinction between flow magnitudes are peak flow versus overall fit [Yapo *et al.*, 1998], low flow versus peak flow versus overall fit [Khu and Madsen, 2005], and low flow versus average flow versus high flow [Tang *et al.*, 2006]. Another example is the work by Boyle *et al.* [2000], who divided the hydrograph into a driven and a non-driven part, based on whether or not there is precipitation in the system.

A common shortcoming of feedforward ANN R–R models is their inability to correctly forecast the timing of peaks, as discussed by de Vos and Rientjes [2005, 2008a]. Since the MSDE penalizes for such timing errors it is likely to be complementary to most other objective functions, which is why it is tested in combination with the MRE. Another combination of two seemingly complementary objective functions is that of the MSLE and the M4E, since they represent the fit on low flows and high flows. The third combination involves all four objectives functions: MSLE, MRE, M4E and MSDE.

A principal difference between MC and SC algorithms is that the former can optimize all objective functions simultaneously while the latter only allows for separate optimization of each objective function. To allow for a comparison of MC and SC results the following approach was taken. For two-objective training, the SC algorithm is run three times: twice for optimization of the two objective functions separately and once where an aggregate of

the two objective functions in the form of their product is taken. The latter is meant to approximate a single optimal trade-off point in Pareto space that values both objective functions equally. For each of the training trials the weights are fixed and for the second objective function the values on training and evaluation data are calculated. In the two-dimensional Pareto plots the combination of both values of both objective functions are subsequently presented (although the training was performed on only one or on the product of two). A three-point approximation of the complete set of Pareto-optimal solutions is hereby generated, allowing a comparison between SC and MC methods.

4.4 Case Study

4.4.1 Data and Models

Data sets from two different river basins have been used in this work (see Appendix A). The first is from the Leaf River basin, located north of Collins, MS, USA. The second data set is from the Geer River basin, which is located in the north of Belgium, North West Europe, and is a subbasin of the river Meuse. Table 4.1 presents descriptions and characteristics of both data sets.

In Chapter 3 the same data set was used and results revealed that ANN model performance drastically improved when a time series of soil moisture was considered as model input. Such a time series reflects the change of soil moisture storage in the catchment by meteorological forcing. A synthetic time series has again been generated using the simple soil moisture reservoir component of the GR4J lumped conceptual R–R model (see Section 3.4.2). Time series of moving averages of rainfall with a window length of 10 days have also been generated for both data sets.

Time series have been split into training and evaluation parts, which share similar statistical features (see Table 4.1). Since the training period contains the largest discharge value, no extrapolation issues were encountered in the present study. Note that these implementations of the evolutionary algorithms were incapable of applying early stopping through cross-evaluation in order to prevent overtraining. Hence, no cross-evaluation procedure was followed for any of the algorithms in order to allow a fair comparison.

Table 4.2 shows the ANN architectures that have developed for the two data sets using the methods described in Section 4.2. ANN1 is applied to the Leaf River basin while ANN2 is applied to the Geer River basin. Because of the larger size of the Leaf River basin it has longer memory and additional input neurons and one extra hidden neuron have consequently been defined. The increased complexity leads to the ANN1 model having twice as many weights as ANN2.

Table 4.1: Data description and statistics. Notes: groundwater measurements are based on one piezometer, soil moisture was simulated using GR4J soil moisture model component, continuity of Geer River time series is largely preserved because the second period starts at the start of the hydrological year and the first period ends at that moment.

Catchment	Period	Variables	Training					Evaluation				
			Min	Max	Mean	St. dev.	Skewness	Min	Max	Mean	St. dev.	Skewness
Geer River (494 km^2)	1980–1991, 1993–1997, daily data	Streamflow (m^3/s)	0.98	14.6	2.49	1.08	2.98	0.96	11.3	2.38	1.07	2.82
		Groundwater (m)	12.4	13.3	13.0	0.24	-0.06	12.4	13.4	13.0	0.30	-0.05
		Areal rainfall (mm)	0.00	138	6.78	12.8	3.19	0.00	221	7.37	14.2	4.20
		Pot. evaporation (mm)	0.00	6.88	1.92	1.50	0.84	0.00	6.80	1.64	1.40	1.02
		Soil moisture (mm)	193	367	306	36.1	-0.78	197	377	314	32.1	-0.87
Leaf River (1944 km^2)	1948–1988, daily data	Streamflow (m^3/s)	1.95	1440	30.0	61.5	7.64	1.56	1310	31.4	68.8	6.54
		Areal rainfall (mm)	0.00	222	3.91	10.3	5.79	0.00	124	3.92	10.1	4.47
		Pot. evaporation (mm)	0.00	8.24	2.85	1.85	0.52	0.00	9.11	2.94	1.92	0.53
		Soil moisture (mm)	58.6	352	220	58.3	-0.22	50.0	358	214	65.1	-0.24

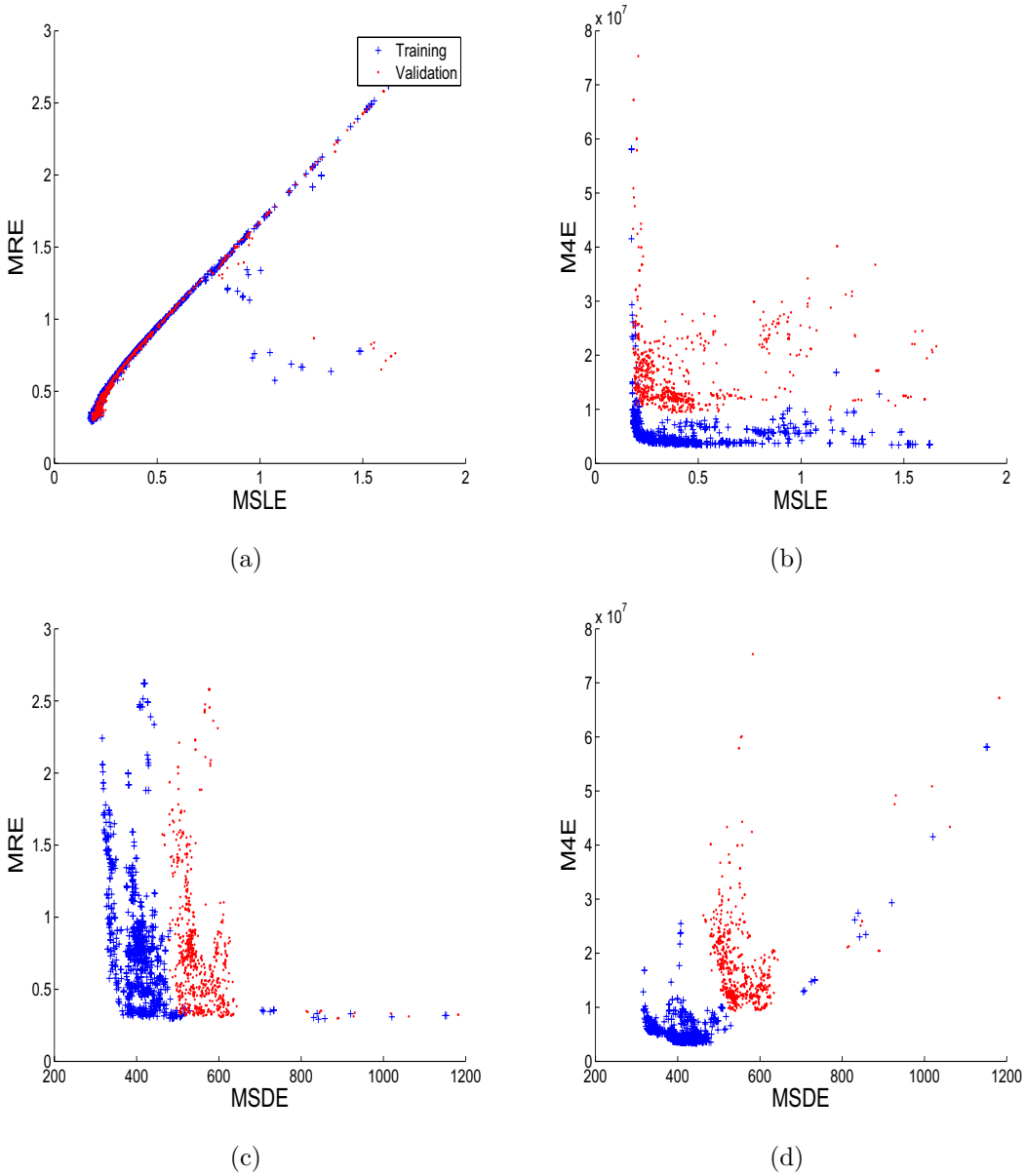


Figure 4.2: Pareto plots of a four-objective optimization run using NSGA–II on Leaf River ANN1 model.

Table 4.2: Description of the two ANN configurations used for simulations. Q is discharge, S is soil water storage, G is groundwater heads, E_P is potential evaporation, $P_{areal,MA}$ is moving average of areal rainfall, P_{areal} is areal rainfall.

Model	Data	Inputs	Time window	Configuration	No. of weights
ANN1	Leaf River	Q	[−1 0]	9–3–1	34
		S	[−1 0]		
		E_P	[−2]		
		$P_{areal,MA}$	[−1 0]		
		P_{areal}	[−1 0]		
ANN2	Geer River	Q	[0]	6–2–1	17
		S	[0]		
		G	[0]		
		E_P	[0]		
		$P_{areal,MA}$	[0]		
		P_{areal}	[0]		

4.4.2 Effects of Choice of Objection Functions

The results of MO training on four objectives (MSLE, MRE, M4E, MSDE) for ANN1 using the NSGA–II algorithm are presented in the two-dimensional projections in Figure 4.2, and they show the trade-off and correlations between the various objective functions. The spread in the solutions in Figures 4.2(c) and 4.2(d) is quite large indicating a significant complexity of a four-dimensional problem. Figure 4.2(c) shows a clear correlation between the MSLE and MRE objective functions, even though they are supposed to represent different hydrograph characteristics. Somehow this difference is ignored by the algorithm and the shape of the four-dimensional front of Pareto solutions is strongly dominated by the trade-offs between the MSDE and the MSLE and MRE functions and trade-off between the MSLE and M4E (i.e., between errors on low flows versus high flows). These results and the fact that the MSDE objective function represents an important indicator of model performance, show that MC training using the MSDE can result in finding a set of important solutions that is often overlooked.

Figure 4.3 shows a scatter plot from the evaluation period for the best solutions found for each of the objectives of a training using NSGA–II on four objectives of ANN1. The figure shows that MRE and MSLE commonly overestimate discharge observations while MSDE commonly underestimates observations. M4E shows small scatter at low flows while scatter increases at higher flows. Overall the scatter plot indicates that the four solutions of the four-dimensional optimization are quite similar, indicating the region in which the algorithm has found its solutions is small.

Numeric results of training ANN1 and ANN2 using various combinations of objective

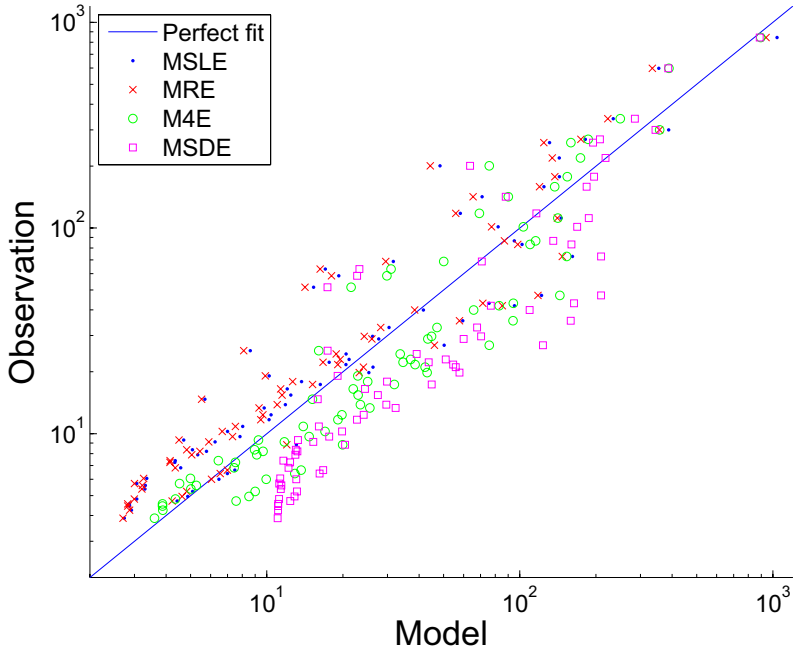


Figure 4.3: Scatter plot for Leaf River ANN1 model showing the single best solutions for each objective function found by a four-objective training run by NSGA–II. Results over one hydrological year from the evaluation period are presented. One out of every five solutions is plotted for improved readability.

functions and algorithms are presented in Tables 4.3 and 4.4 respectively. Note that the mean and standard deviations apply to the best 80% of solutions found by the algorithm, and individual solutions can still have higher or lower values for any of the objective functions. This threshold of 80% was arbitrarily chosen to exclude poorly performing solutions without disregarding many solutions. The results in Tables 4.3 and 4.4 are considered a representation of the location and size of the region in which the algorithm finds its well-performing solutions.

Most combinations of MRE and MSDE functions have higher accuracy than the training on MRE alone while the spread is also smaller. This indicates that the addition of the MSDE function constrains the optimization to a smaller and better solution region, thereby indicating the effectiveness of the function in ANN R–R training. Results featuring the MSLE and M4E functions show that including more objective functions not necessarily improves the quality of the training results. This is most obviously seen in the results for ANN1, whereas ANN2 generally still improves by considering more objective functions. It is assumed that this is most likely due to a strong trade-off between the various objective functions (most notably the M4E). The nature of this particular combination of model, data and objective functions results in a solution space with multiple regions of attraction, and the effectiveness of the training algorithm becomes very determining for the quality of the training procedure. In this light, it is highly interesting that the NSGA–II algorithm performs best on ANN1 when all objective functions are used. Apparently, this algorithm is the only one able to deal with this complex solution space. In summary, these results are an indication that the inclusion of multiple appropriate objective functions can result in more reliable training of ANN models.

4.4.3 Performance of Training Algorithms

Tables 4.3 and 4.4 also allow comparison of the various algorithms and indicate that LM is very powerful and often has the highest accuracy on most objective functions for both ANN catchment models. Nevertheless, it is commonly outperformed by other algorithms on the MSDE function. The GA may be considered the poorest performer and has difficulty with optimizing the M4E function. NSGA–II outperforms MOSCEM–UA on the Leaf River model (ANN1) but the two produce very similar results for the Geer model (ANN2).

The above is shown in more detail in Figures 4.4 and 4.5, which show objective space plots for combinations of objective functions, optimization algorithms, for ANN1 and ANN2 respectively. The criteria to compare algorithm performance from these figures are (1) the closeness of the solutions to the origin (i.e. accuracy), (2) the similarity of the shape and location of the evaluation results compared to the training results, and (3) low spread of results. The latter two specifically indicate consistency and reliability. Following the

Table 4.3: Evaluation results of the Leaf River ANN1 model trained with various algorithms on several combinations of objective functions. Performance is expressed using six objective functions with means and standard deviations. As a visual aid, the best values for each performance measure are shaded for every algorithm.

Algorithm	Obj. functions	MRE	C_{NS}	C_{PI}	MSLE	M4E ($\cdot 10^6$)	MSDE
LM	MRE	0.28±0.08	0.93±0.00	0.70±0.02	0.18±0.12	9.0±6.7	629±39
	MRE·MSDE	0.29±0.07	0.93±0.00	0.70±0.01	0.20±0.12	6.0±1.6	645±35
	M4E	0.60±0.27	0.93±0.01	0.68±0.03	0.47±0.23	5.3±1.3	660±31
	MSLE·M4E	0.62±0.21	0.90±0.03	0.56±0.12	0.49±0.21	10.6±7.5	701±52
	MSLE·MRE·M4E·MSDE	0.50±0.19	0.92±0.02	0.65±0.08	0.39±0.23	7.9±6.7	683±63
GA	MRE	1.79±1.09	0.65±0.14	-0.60±0.65	1.39±0.75	115±91	1070±620
	MRE·MSDE	1.52±0.78	0.80±0.02	0.07±0.11	0.95±0.36	39.4±30.5	576±54
	M4E	2.58±0.77	0.73±0.07	-0.26±0.33	1.49±0.40	40.1±25.2	974±290
	MSLE·M4E	2.97±1.35	-2.30±2.08	-14.1±9.57	1.57±0.63	160±203	5640±5240
	MSLE·MRE·M4E·MSDE	2.20±0.96	0.16±0.52	-2.85±2.37	1.38±0.53	740±625	1920±1580
NSGA–II	MRE, MSDE	0.84±0.41	0.74±0.01	-0.18±0.05	0.69±0.53	215±6.0	717±24
	MSLE, M4E	0.94±0.65	0.62±0.11	-0.72±0.52	0.63±0.35	342±563	1750±600
	MSLE, MRE, M4E, MSDE	0.65±0.28	0.84±0.01	0.26±0.05	0.34±0.15	14.2±3.9	560±47
MOSCEM–UA	MRE, MSDE	2.31±0.69	0.83±0.03	0.26±0.13	1.22±0.32	9.85±4.83	555±88
	MSLE ,M4E	1.34±0.52	0.89±0.01	0.52±0.04	0.79±0.22	7.14±0.00	746±39
	MSLE, MRE, M4E, MSDE	3.45±2.90	0.29±0.24	-2.24±1.09	2.04±1.17	313±2.01	1040±226

Table 4.4: Evaluation results of the Geer River ANN2 model trained with various algorithms on several combinations of objective functions. Performance is expressed using six objective functions with means and standard deviations. As a visual aid, the best values for each performance measure are shaded for every algorithm.

Algorithm	Obj. functions	MRE	C_{NS}	C_{PI}	MSLE ($\cdot 10^{-2}$)	M4E	MSDE
LM	MRE	0.14±0.03	0.81±0.03	0.63±0.06	2.94±0.87	0.58±0.11	0.27±0.01
	MRE·MSDE	0.17±0.02	0.78±0.02	0.58±0.04	3.74±0.70	0.60±0.12	0.26±0.02
	MSLE·MRE·M4E·MSDE	0.13±0.03	0.83±0.02	0.66±0.05	2.44±0.71	0.57±0.12	0.27±0.01
GA	MRE	0.27±0.02	0.47±0.11	-0.03±0.22	8.46±1.20	4.63±4.48	0.49±0.17
	MRE·MSDE	0.27±0.02	0.57±0.03	0.16±0.06	7.91±0.96	1.36±0.31	0.21±0.02
	MSLE·MRE·M4E·MSDE	0.21±0.02	0.62±0.06	0.26±0.12	5.54±0.82	2.42±1.38	0.28±0.05
NSGA–II	MRE, MSDE	0.23±0.02	0.67±0.05	0.35±0.01	6.13±1.07	1.02±0.10	0.23±0.00
	MSLE, MRE, M4E, MSDE	0.20±0.02	0.68±0.03	0.37±0.05	5.66±0.70	1.61±0.66	0.28±0.04
MOSCEM–UA	MRE, MSDE	0.23±0.00	0.66±0.00	0.34±0.00	6.37±0.22	0.79±0.06	0.21±0.01
	MSLE, MRE, M4E, MSDE	0.15±0.00	0.68±0.00	0.38±0.00	4.04±0.16	2.49±0.16	0.28±0.06

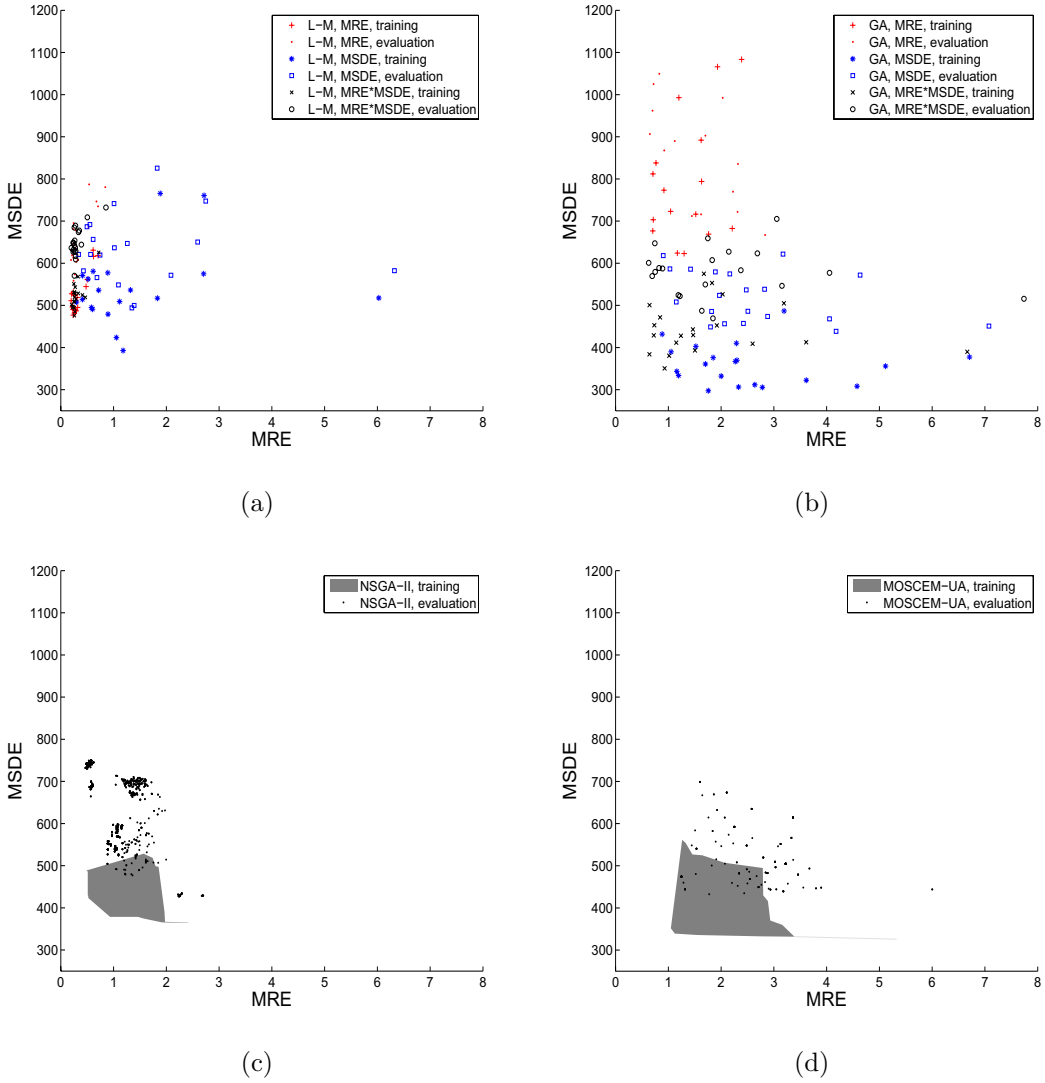


Figure 4.4: Pareto plots of Leaf River ANN1 model performance after being trained on the MRE and MSDE objective functions using single-criterion (a and b) and multi-criteria (c and d) algorithms.

description in Section 4.3.3, for SC optimization the figures show three training trials that together approximate the front of Pareto solutions Note there are few solutions outside the bounds of the plots where an algorithm got stuck in a very poor-performing local optimum.

Figure 4.4 show objective space plots on the performance of model ANN1 after training using, respectively, the LM, GA, NSGA–II and MOSCEM–UA algorithms on the MRE and MSDE objective functions. The LM algorithm generally finds small solution regions that are close to the origin, but no clear Pareto front is discernible since the algorithm has some difficulty in optimizing the MSDE function. Even with the MRE·MSDE objective function the MSDE is basically ignored, judging from the similarity with MRE training. The nature of the MSDE likely causes the response surface of this objective function to be very irregular and this causes the gradient-based LM to get stuck in local minima. Figure 4.4(b) shows that the GA optimizes the MSDE function better than the LM although solutions are often quite far from the origin and have large spread. This suggests an inability of the GA to fine-tune its solutions to higher accuracy, a problem that is common in GA optimization. The grey areas in Figures 4.5(c) and 4.5(d) delimit the region in which the Pareto solutions fall that were found after 10 training runs with the MC algorithms. This way, a region of Pareto solutions is found that is more representative of algorithm performance than a single front of Pareto solutions found after a training run. The dots represent results from evaluation. The MOSCEM–UA has many duplicate solutions which is why fewer dots than NSGA–II are plotted in both Figures 4.4 and 4.5. The solutions indicate a trade-off between the MRE and MSDE although the spread is low as opposed to the LM and GA results. The Figures 4.4(c) and 4.4(d) show that the NSGA–II and MOSCEM–UA perform similar on the training data. The latter shows slightly more consistency since the evaluation results are very similar to the training results in terms of location and shape. A comparison between the subfigures of Figure 4.4 proves that both MC algorithms find a better set of trade-off solutions than the SC algorithms, even when the product of both functions is taken in the latter case. The MC algorithms also show to be more consistent in terms of both objective functions since both training and evaluation solutions fall within a relatively small region.

Likewise, simulations were done for the more parsimonious ANN2 with the MSLE and M4E functions for low and high flow, the results of which are shown in Figure 4.5. It seems that LM again is quite consistent in finding accurate solutions in a small region. However, a significant number of results are scattered in low accuracy regions, indicating the algorithm gets stuck in local optima. Moreover, it is difficult to discern a clear front of Pareto solutions which is unexpected given the theoretical trade-off between high-flow and low-flow fit. The training with the MSLE·M4E objective function, on the other hand, seems to give quite accurate and consistent results, proving that even a SC algorithm is able to benefit from using multiple objectives in some way. LM generally outperforms the

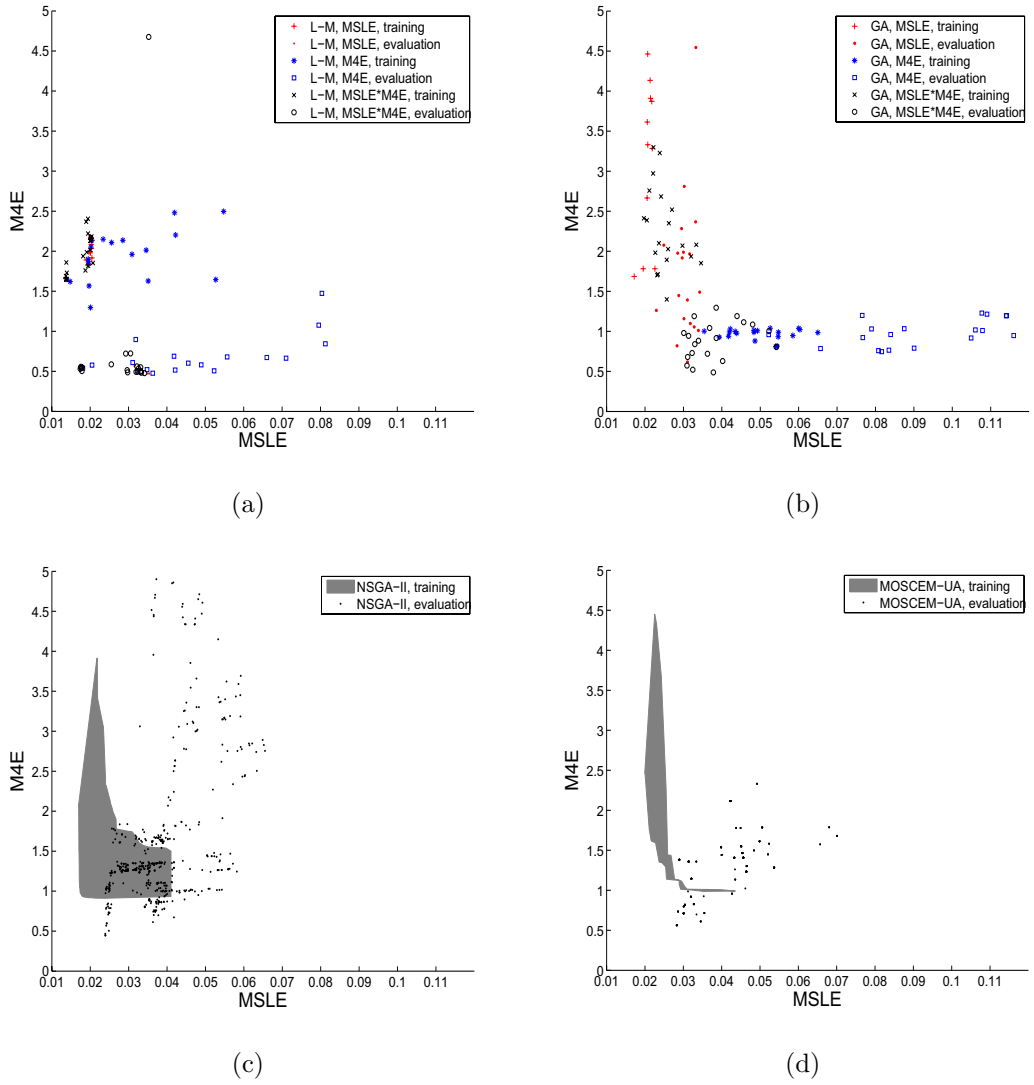


Figure 4.5: Pareto plots of Geer River ANN2 model performance after being trained on the MSLE and M4E objective functions using single-criterion (a and b) and multi-criteria (c and d) algorithms.

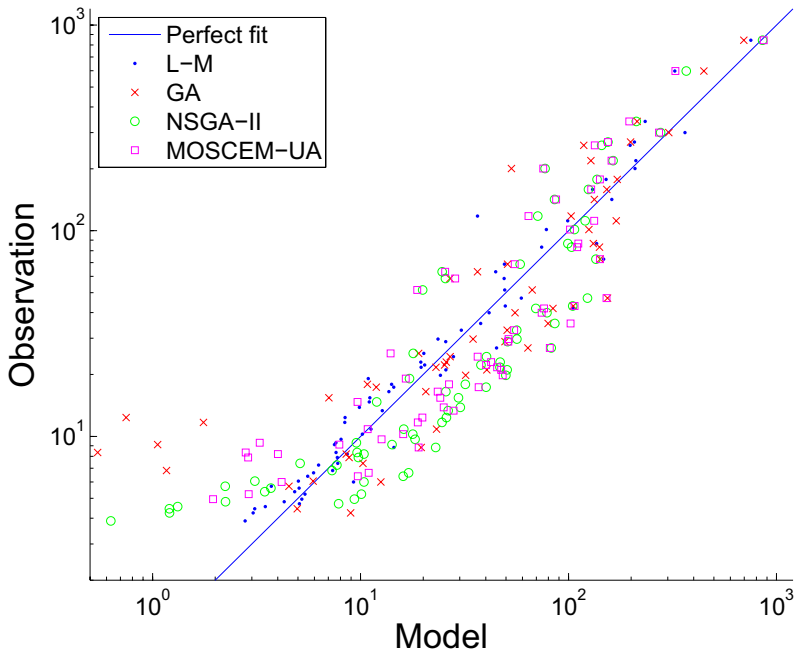


Figure 4.6: Scatter plots for Leaf River ANN1 model showing the best solutions for MRE found by various algorithms. Results over one hydrological year from the evaluation period are presented. One out of every five solutions is plotted for improved readability.

GA in a similar way to what is shown in Figure 4.4. The GA shows a clear trade-off in its Pareto solutions and has no outliers in bad-performing regions. It therefore seems to search in the right region, but is not able to consistently produce accurate results judging from the large spread in results.

When comparing the MC results in Figure 4.5(c) and 4.5(d) it can be observed that the NSGA–II often finds Pareto solutions that are closer to the origin, indicating higher accuracy (the left side of the gray polygon). Although the size of the gray area is large due to some less accurate training runs, and while the evaluation results do not show a clear trade-off, the majority of NSGA–II solutions has high accuracy and low spread. NSGA–II, however, seems to give slightly less satisfying evaluation results. The MOSCEM–UA is more consistent judging from its narrow Pareto region and the shape of the evaluation results, but is slightly less accurate judging from the distance to the origin.

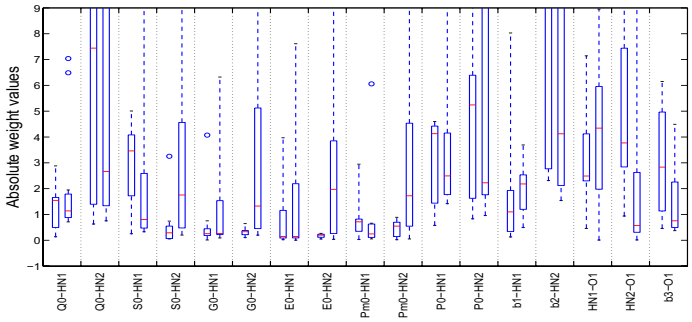
Figure 4.6 shows a scatter plot with the best solutions found by the four different algorithms for the MRE objective function (the MC solutions were taken from the MRE versus MSDE training results). The differences between the results of the various algorithms are often just as large as those between the results for the various objective functions. The latter, however, show more consistency, whereas the former are noisier. Nevertheless, this suggests that the performance of ANN models can hinge just as much on the choice of training algorithm as on the choice of objective function(s).

4.4.4 Weight Analysis

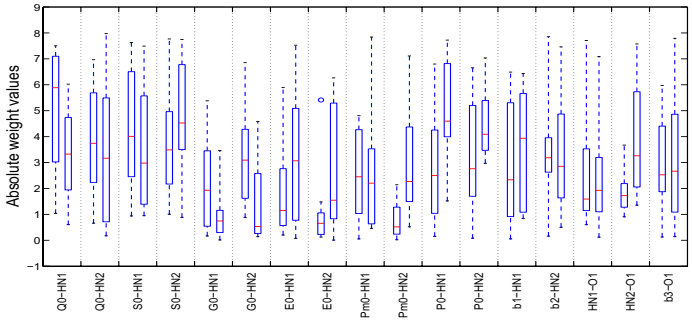
The absolute values of the posterior weight distribution of the Geer River ANN2 model trained on MSLE and M4E using SC and MC training algorithms are displayed in the box-and-whisker plots of Figure 4.7. The weights on the connections between the various inputs (see Table 4.2) and the two hidden neurons (indicated as HN1 and HN2) are displayed in the first 12 columns. The next two columns show the weights on the bias signals to the hidden neurons. The last three columns show the weights on the signals to the output neuron from the two hidden neurons and the bias signal, respectively. Each column contains two bars, which show the distribution for the best 20 solutions according to the MSLE (left bar) and the M4E (right bar) of the training period.

The relative contribution of each of the input variables follows from the absolute values of the weights. The previous discharge, soil moisture and precipitation often dominate, whereas the groundwater and evaporation input variables generally gets assigned small weights, thereby limiting their influence. Another interesting observation from Figure 4.7 is that there are significant differences between the optimized weight distributions found by the various algorithms. The LM algorithm shows very large values and spread in values for some of its variables, indicating that it often finds different solutions for each training run. The GA has significant spread as well, and seems to have difficulty in deciding which input variable should be assigned the biggest weights. MOSCEM–UA and NSGA–II to some degree seem comparable in both their values and in their spread of solutions. On the one hand, this is not surprising considering the resemblances in terms of objectives between these algorithms (see Figures 4.4 and 4.5). On the other hand, the algorithms work differently and the fact that both end up in the same solution region indicates that this is a stable region of attraction.

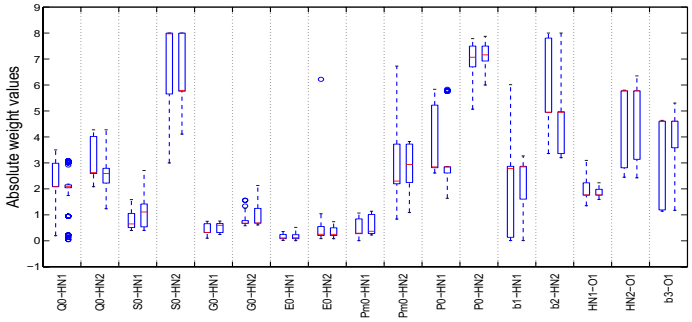
When comparing weights of SC to MC optimization, results show that spread in the optimized weight distribution is largest for the SC algorithms. SC optimization also shows large changes in sensitivity toward specific inputs while in MC optimization only a small number of inputs have significant effect whereas other inputs have relatively small effect. When reviewing the results from Figures 4.6, this difference in sensitivity between algorithms is reflected in the spread of results in the solution region. As such, the MC algorithms seem to be more consistent and stable in their optimization, since only a few relevant inputs appear to be sensitive and weight estimates are found in a more consistent manner. Training on other objective functions (not shown here) showed similar differences in spread between the algorithms.



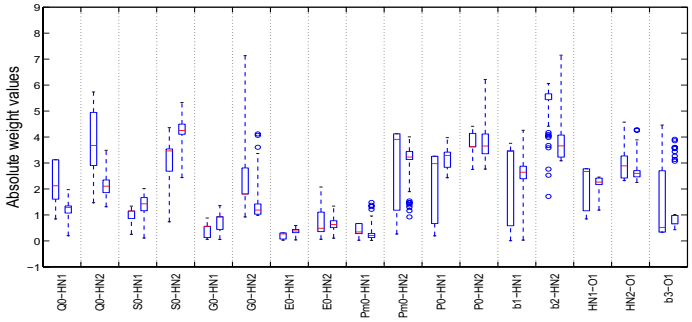
(a)



(b)



(c)



(d)

Figure 4.7: Posterior weight distributions of training results the ANN2 model. Figure (a), (b), (c) and (d) show the LM, GA, NSGA–II and MOSCEM–UA results, respectively. The 17 columns represent the 17 ANN weights and the two bars in each plot the best 20 solutions for the MSLE (left boxes) and M4E (right boxes). The boxes depict the median and upper and lower quartiles. The whiskers show the most extreme values within 1.5 times the interquartile range from the ends of the box. Circles are outliers.

4.5 Summary and Discussion

Similar to MC calibration of conceptual hydrological models, trade-offs between objective functions also manifest themselves in MC ANN training. By constraining the search for optimal ANN weights by using MC training, solutions were found that offer a good compromise in performance on multiple objectives. In this research it is also shown that by using multiple objectives more information can be extracted from the data. Results indicate that by using MC training, more stable regions in the weight space can be identified which result in more reliable models compared to SC training.

However, the comparison of the LM, GA, MOSCEM–UA and NSGA–II algorithms shows that they all have their respective pros and cons. The LM algorithm often gives accurate results but does not produce very consistent weights, suggesting low reliability. The GA appears to find reasonably well-performing regions in the weight space but is often unable to fine-tune to good optima, making it the poorest performer of the algorithms tested. The MOSCEM–UA and NSGA–II algorithms find solutions that are commonly better than the LM and GA algorithms. Specifically, they are able to consistently locate specific regions in the weight space in which good solutions can be found for several objective functions. Nevertheless, most algorithms show significant spread in results and sometimes even inconsistency in the performance and the *a posteriori* weight distributions of the ANNs. This suggests that future research should consider that ANN performance can have significant uncertainty due to inadequacies in optimization algorithms. Clearly, there is a need to use sophisticated methods for optimization algorithms in ANN training. A possible alternative to the four algorithms presented in this chapter are so-called memetic algorithms [Hart *et al.*, 2005], which combine global and local search strategies. Another alternative that is able to combine the strengths of individual algorithms is the AMALGAM multialgorithm by [Vrugt and Robinson, 2007].

Additionally, in most of the examples presented in this chapter, the clear trade-off between the MSDE and the traditional objective functions indicates that the MSDE objective function exploits information that is usually ignored in hydrological model calibration. Since MSDE penalizes for hydrograph shape errors, especially timing errors and noise, it can be argued that this objective function can provide valuable information in model calibration. Clearly, it is difficult and precarious to generalize beyond the results presented here and more research on these issues is needed.

Chapter 5

Multi-Criteria Comparison of Artificial Neural Network and Conceptual Rainfall–Runoff Models

Modified from:

de Vos, N. J., Rientjes, T. H. M., 2007. Multi-objective performance comparison of an artificial neural network and a conceptual rainfall–runoff model. *Hydrol. Sci. J.* **52**(3), 397–413.

Abstract

This chapter presents a multi-criteria comparison between an artificial neural network and the conceptual HBV R–R model. The popular NSGA–II algorithm was used for calibration of both models. A combination of three objective functions was used to evaluate model performance. The results show that, for a small forecast lead time, the artificial neural network outperformed the HBV model on the objective functions for low and high flows, but the former was outperformed on a objective function related to the shape of the hydrograph. As the forecast horizon increases, the HBV model more and more outperforms the ANN model on all objective functions. The main conclusion of this chapter is that, although the differences between the two model approaches make a straightforward and unequivocal comparison difficult, the multi-criteria approach enables a more reliable evaluation of the two models than the single-objective approach.

5.1 Introduction

The most popular models for R–R modeling are conceptual models, which are based on the principle of mass conservation and simplified forms of momentum and energy conservation principles, as discussed in Chapter 2. Alternatively, data-driven modeling approaches such as ANN models can be used. The usefulness of ANNs in runoff forecasting has been researched extensively over the last decade (see Chapter 2), but the modeling community is far from reaching a consensus on the matter.

One of the drawbacks of ANNs is their apparent lack of physical interpretability. The empirical and black box nature of the models raises doubts about the consistency of the model with real-world processes and catchment characteristics. Do ANNs produce their-often reasonably accurate-results for the right reasons? The previous chapters suggest a multi-criteria viewpoint on the performance evaluation of ANNs. One of the issues that has been explicitly addressed in this work is the timing errors that are frequently made by ANNs, and the apparent trade-off between these timing errors and the overall fit of the predictions (see also Abrahart *et al.* [2006]).

In recent years, several studies have compared ANNs and conceptual models (e.g., Hsu *et al.* [1995]; Dibike and Solomatine [2001]; Tokar and Markus [2000]; Gaume and Gosset [2003]). However, there is a clear lack of studies on the subject of comparisons between ANNs and conceptual models using multiple criteria. Given the fact that the calibration problem inherently involves multiple criteria [Gupta *et al.*, 1998], the question arises whether a multi-criteria view can shed new light on the performance differences between ANNs and conceptual models.

The aim of this chapter is to quantitatively and qualitatively compare the performance of an ANN and a conceptual R–R model for a meso-scale catchment in terms of multiple criteria. Moreover, the trade-offs between various objective functions for the two types of R–R models are investigated. The above is accomplished by calibrating a feed-forward ANN model and the HBV conceptual model using the NSGA–II multi-criteria optimization algorithm. Two different forecast horizons (i.e. 1 and 6 hours) are used for both models.

5.2 Model Descriptions

5.2.1 Artificial Neural Network Model

The same ANN configuration was used as in Chapter 4. Figure 4.1 shows an example of this type of ANN.

An analysis of linear correlation coefficients and the nonlinear average mutual information (AMI) [Gallagher, 1968] between the discharge time series and various other time series served as indication for the usefulness of certain variables as ANN input (see Table 5.1). The AMI is based on Shannon's theory of entropy [Shannon, 1948] and is defined as the average of the mutual information:

$$I(\mathbf{x};\mathbf{y}) = H(\mathbf{x}) - H(\mathbf{x}|\mathbf{y}) \tag{5.1}$$

where $H(\mathbf{x})$ is the entropy of $\mathbf{x}$, which is a measure of its uncertainty, and $H(\mathbf{x}|\mathbf{y})$ the conditional entropy that expresses the information in $\mathbf{x}$ given that $\mathbf{y}$ is known. In an

Table 5.1: ANN model input variables and time windows for the two different forecast horizons. Note that only the rainfall input time window is different for the two models.

Lead time	Input variables				
	Rainfall	Mov. avg. rainfall	Soil moisture	Previous discharge	Mov. avg. discharge
1 hour	t-11 to t-7	t0	t0	t-2 to t0	t0
6 hours	t-6 to t-2	t0	t0	t-2 to t0	t0

equivalent but more practical formulation it can be seen how the AMI can be calculated using the joint and marginal probability distribution functions of $\mathbf{x}$ and $\mathbf{y}$:

$$I(\mathbf{x};\mathbf{y}) = \sum_{y\in\mathbf{y}}\sum_{x\in\mathbf{x}} p(x,y)\cdot\log_2\left(\frac{p(x,y)}{p(x)\,p(y)}\right) \tag{5.2}$$

High values of the AMI served as primary criteria for input selection.

By gradually adding neurons until the ANN's mean squared error no longer decreased significantly, an appropriate number of hidden neurons was found to be 3. The network also contained biases, so the 11–3–1 network contained 40 parameters that had to be calibrated. Details on the calibration procedure can be found in Section 5.3.

5.2.2 HBV Model

The HBV conceptual R–R model, originally developed by Bergström [1976], has been applied in a large number of countries and over a large range of hydrological conditions. A re-evaluation of the model with various modifications and additions, termed HBV–96, is presented in Lindström *et al.* [1997]. The reader is referred to that paper for a detailed review and analysis of the model.

In this study, a simplified version of the HBV–96 model is used. A lumped model structure without a snow routine was transformed to be used for simulations with hourly time steps, whereas the model is normally operated on daily time steps. Figure 5.1 shows a schematic of the HBV model structure.

The HBV model has three state variables: soil moisture (S), upper zone storage (U), and lower zone storage (L). The inputs to the HBV model are rainfall (P) and potential evaporation (E_p). The ρ parameter is multiplied with the rainfall to calculate the amount of infiltration that will be added to S:

$$I = \rho \cdot P \tag{5.3}$$

The actual evaporation that is subtracted from S is calculated as

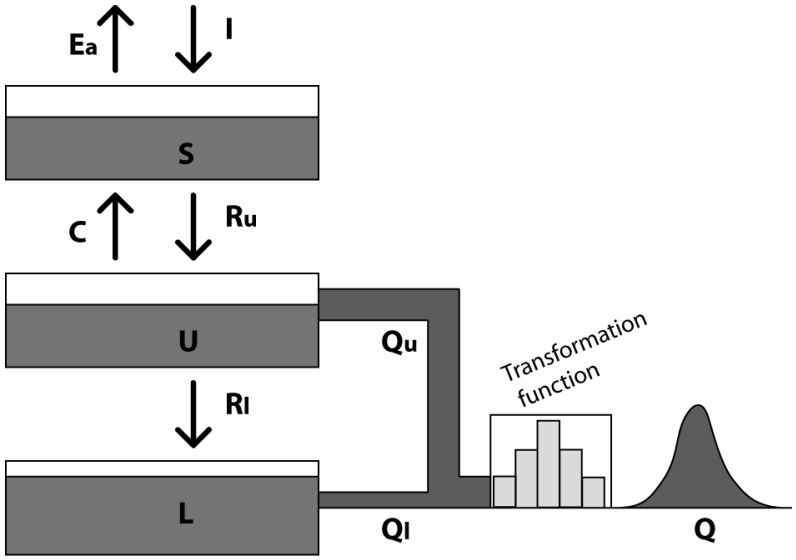


Figure 5.1: Schematic of the HBV model structure.

$$E_a = \begin{cases} E_p \cdot (S/E_{pl}), & \text{if } S < E_{pl} \\ E_p, & \text{if } S \geq E_{pl} \end{cases} \tag{5.4}$$

Capillary flux from the upper zone to the soil moisture zone is calculated as

$$C = C_{\max} \cdot (1 - S/S_{\max}) \tag{5.5}$$

and the recharge from the soil moisture zone to the upper zone as

$$R_u = I \cdot (S/S_{\max})^{\beta} \tag{5.6}$$

The percolation from the upper to the lower zone (R_l) is a constant value, which is calibrated. Subsequently, the response from U and L can be determined.

$$Q_u = k_u \cdot U^{1+\alpha} \tag{5.7}$$

$$Q_l = k_l \cdot L \tag{5.8}$$

Finally, the sum of Q_u and Q_l is transformed through a triangular transformation function with base length t_f to get the discharge Q.

Table 5.2 shows a description of the various calibration parameters of the HBV model, along with the ranges within which the parameters were calibrated. Different calibrated values of the parameters of the HBV model were used for the one-hour-ahead and the six-hour-ahead forecasts.

Table 5.2: Description of HBV model parameters

Name	Description and unit	Prior range
ρ	Rainfall correction factor [-]	0–1
S_{max}	Maximum soil moisture content [L]	0–800
E_{pl}	Limit for potential evaporation [LT^{-1}]	0–1
β	Parameter in soil routine [-]	1–2
C_{max}	Maximum value of capillary flow [LT^{-1}]	0–0.2
k_u	Recession coefficient upper zone [T^{-1}]	0–0.1
α	Response box parameter [-]	0–1
R_l	Percolation [LT^{-1}]	0–0.5
k_l	Recession coefficient lower zone [T^{-1}]	0–0.01
t_f	Transformation function parameter [T]	1–16

5.3 Multi-Criteria Calibration Approach

5.3.1 NSGA–II Algorithm Settings

The NSGA–II algorithm was chosen as optimization algorithm (explained in Section 4.3.2). For the ANN calibration, the population size was chosen to be 80, and the number of generations 800 (resulting in 64,000 model evaluations). The conceptual model has fewer parameters, so a population of 40 was found to be sufficient, along with a number of 400 generations (resulting in 16,000 model evaluations). The ANN model structure has 40 weights and the HBV 10 parameters, justifying the difference between the numbers of model evaluations. Other parameters that are kept constant for all calibration procedures are the probabilities of crossover and mutation, which are set to 0.9 and 0.05 respectively, and the crossover and mutation distribution indices, which are both set to 20. These values were found by testing some commonly suggested values for these settings (e.g., Deb [2001]).

Both the ANNs and the conceptual models are initialized by generating normally distributed random values for the parameters between reasonable ranges. It is partly because of this randomness that the optimization algorithms can find different optima in the objective function response surface for each new calibration trial. Because of their global perspective, however, methods such as GA are theoretically able to find global optima with a high probability [Duan *et al.*, 1992; Goldberg, 2000; Deb, 2001]. Here it is therefore assumed that the NSGA–II algorithm is able to find acceptable optima and the dependency on initial values is neglected. This assumption also seems warranted by the results in Chapter 4.

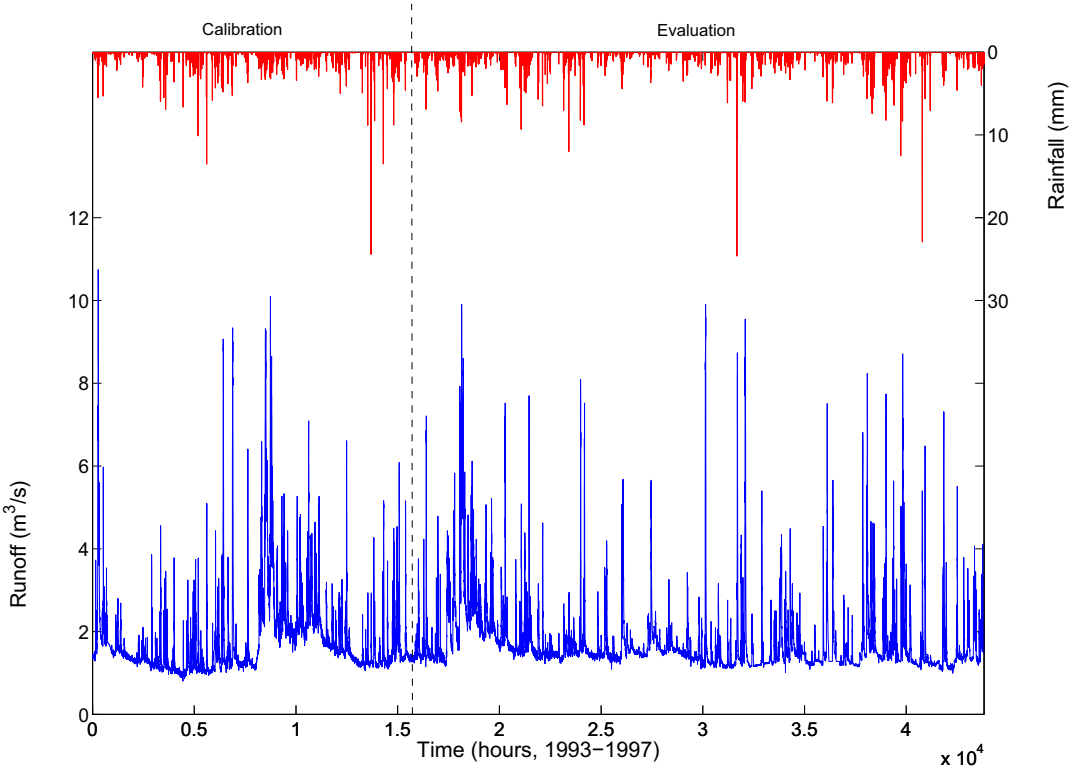


Figure 5.2: Time series of discharge at Kanne and rainfall at Bierset. The time series are split into calibration and evaluation parts.

5.3.2 Objective Functions

Each of the objective functions used in this study is intended to cover a unique aspect of the streamflow hydrograph. Unfortunately, correlations between them are inevitable and the choice of which objective functions to use remains subjective. The objective functions that were used in this study — Mean Squared Error, MSE; Mean Squared Logarithmic Error, MSLE; Mean Squared Derivative Error, MSDE — are discussed in Chapters 3 and 4.

5.4 Case Study

5.4.1 Selected Data

The hourly Geer River basin data set were used for this research (see Appendix A). Figure 5.2 shows the hourly catchment discharge in combination with rainfall at Bierset for the complete data period.

The simple soil moisture reservoir component of the GR4J lumped conceptual R–R model [Edijatno *et al.*, 1999; Perrin *et al.*, 2003] was again used to produce a time series of

Table 5.3: Descriptive statistics for the discharge data.

	Min.	Max.	Mean	Std. dev.	Skewness	Kurtosis
Calibration	0.80	10.74	1.80	1.04	3.60	20.3
Evaluation	1.00	9.90	1.74	0.86	4.11	25.2

simulated soil moisture (S, see Figure 3.15). Time series of the non-decaying moving average of the discharge and the rainfall were also constructed. These procedures are identical to the ones presented in Chapters 3 and 4.

The time series were split into calibration and evaluation periods (see Fig. 5.2), which shared similar statistical features, as shown in Table 5.3. The calibration period contained the largest discharge value, so no extrapolation issues were encountered in the present study. No measure for preventing overfitting of the ANN models was used since the use of a training algorithm based on evolutionary principles generally produces sub-optimal solutions, thereby significantly reducing the risk of overfitting [Dawson *et al.*, 2006]. Moreover, the algorithm iterations were limited to a reasonably small number and the results were inspected afterward to check for possible overtraining effects.

5.4.2 Results

Figure 5.3(a) shows Pareto plots of the three objectives for the calibration results of the one-hour-ahead and six-hour-ahead forecasts of both the ANN and the HBV models. Figure 5.3(b) shows the accompanying evaluation results. The similarity between the calibration and evaluation results for both models suggests that the calibration runs found solutions in good regions of the parameter space that are capable of generalizing. The results also clearly indicate a correlation between the MSLE and MSE in proportion to the trade-off between the MSDE and these two functions. Apparently, with the MSDE function different types of solutions are valued in comparison with the MSE and MSLE. The MSDE function is physically interpretable as a measure for the shape of the hydrograph that especially penalizes timing errors and noisy approximations. From this it is concluded that using the MSDE in a multi-criteria calibration procedure results in finding a more diverse set of physically realistic model realizations than if the more common combination of MSE and MSLE was used. This conclusion was also reached in Chapter 4.

The ANN model has a higher-dimensional parameter space compared to the HBV model, which is why a more extensive calibration routine was performed (see Section 5.3). This results in a more diverse set of solutions than for the HBV model, encompassing more extreme solutions for all of the objective functions. On average, however, the ANN seems to often outperform the HBV model in terms of the MSE and MSLE objective functions,

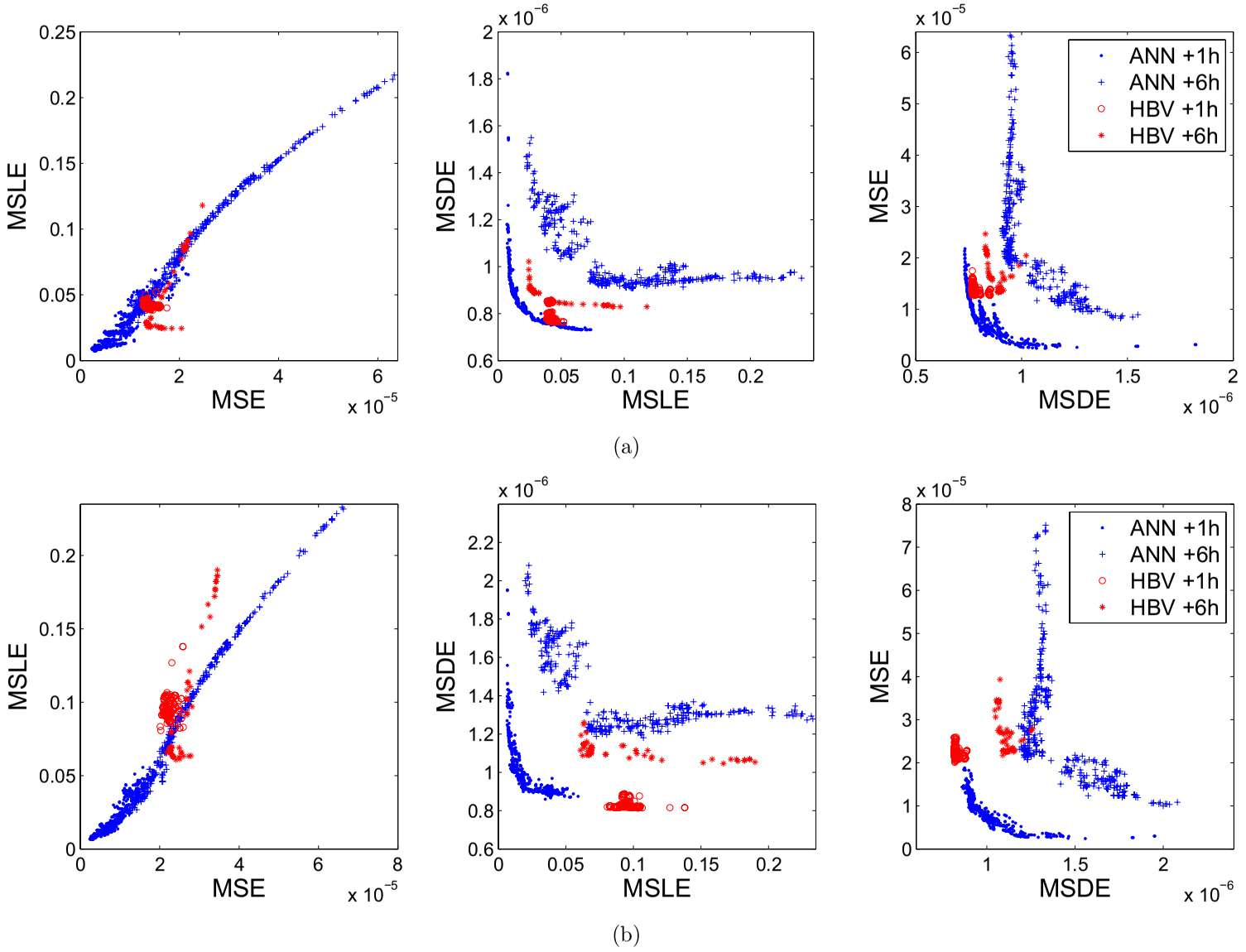


Figure 5.3: Pareto plots showing (a) calibration and (b) evaluation results for both the ANN and the HBV models.

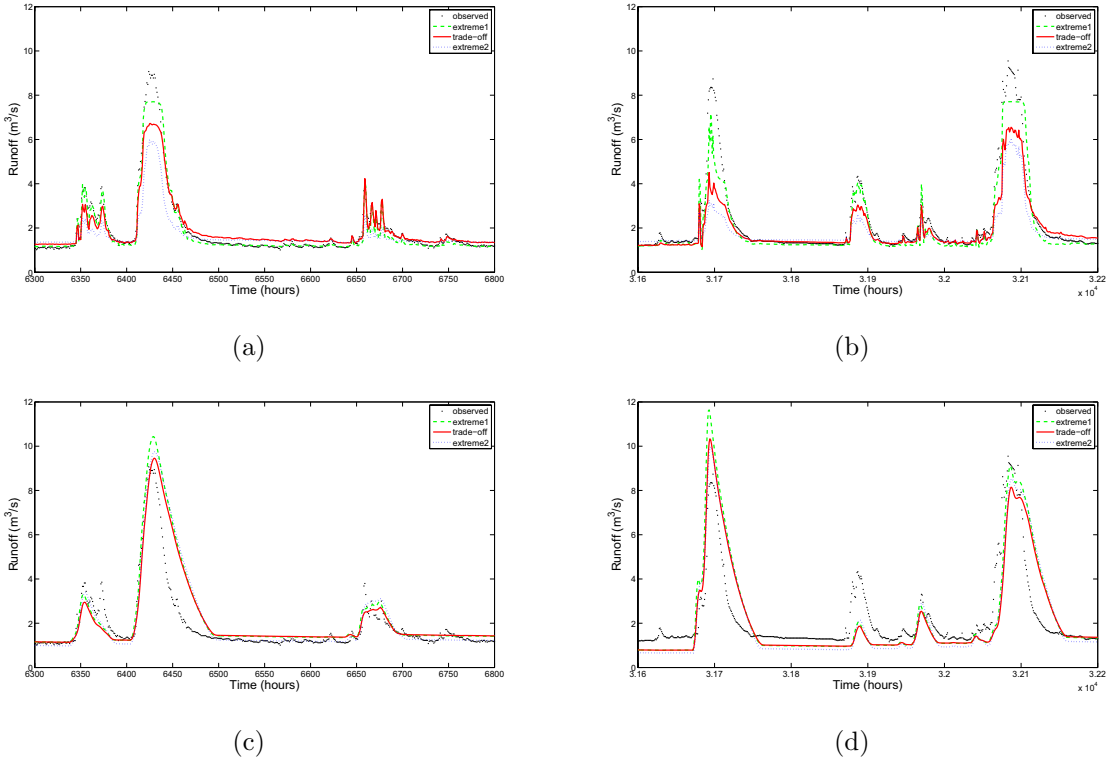


Figure 5.4: Simulation details for one-hour-ahead forecasts of the ANN (a,b) and HBV (c,d) models. Calibration (a,c) and evaluation (b,d) details are plotted for two extreme solutions and one trade-off solution from each calibration Pareto front. Table 5.4 specifies the location of each of these plotted solutions on the Pareto front.

while the opposite is true for the MSDE function. A comparison between the one-hour-ahead and six-hour-ahead simulations shows that the HBV model performs better relative to the ANN model as the forecast lead time increases. This indicates that the physical principles underlying the HBV (e.g. the mass balance equation) help the model to maintain forecasting capabilities for larger forecast horizons, while the ANN model as a data-driven technique has increased difficulty in extracting the rainfall-runoff transformation from the data.

These fundamental differences between the ANN and HBV are also clearly recognized in the simulations shown in Fig. 5.4. These plots show calibration and evaluation details of three Pareto solutions for the one-hour-ahead forecast of both models. One of the Pareto solutions represents a trade-off between the three objectives. The two others were chosen from the extremes of the Pareto front. Table 5.4 shows the coordinates of the solutions in Pareto space.

Some general observations are that the ANN generally underestimates high peak flows but performs well on low flows. The HBV, on the other hand, performs poorly on lower flows and shows an inability to accurately simulate the recession limb of the hydrograph. The

Table 5.4: Coordinates in Pareto space of the trade-off and extreme solutions plotted in Fig. 5.4. The bold numbers indicate low values and hence good extremes for the various objective functions.

		Objective functions		
Model	Solution	MSE	MSLE	MSDE
ANN	Trade-off	$1.486 \cdot 10^{-6}$	$0.629 \cdot 10^{-1}$	$1.230 \cdot 10^{-6}$
	Extreme 1	$0.914 \cdot 10^{-6}$	$0.202 \cdot 10^{-1}$	$1.479 \cdot 10^{-6}$
	Extreme 2	$6.101 \cdot 10^{-6}$	$2.082 \cdot 10^{-1}$	$0.934 \cdot 10^{-6}$
HBV	Trade-off	$1.325 \cdot 10^{-6}$	$0.405 \cdot 10^{-1}$	$0.785 \cdot 10^{-6}$
	Extreme 1	$1.530 \cdot 10^{-6}$	$0.521 \cdot 10^{-1}$	$0.765 \cdot 10^{-6}$
	Extreme 2	$1.387 \cdot 10^{-6}$	$0.391 \cdot 10^{-1}$	$0.854 \cdot 10^{-6}$

extreme solutions show that a low MSDE sometimes leads to very poor performance in terms of magnitude, especially for the ANN. The poorer ANN performance on the MSDE seems to be largely due to noisy model output and to timing errors of ANN models. The high autocorrelation in the hourly runoff time series, in combination with the fact that the last known runoff value is used as an input to the ANN, results in the ANN presenting an output that is very similar to the last known runoff input. This effectively results in a timing error of the forecast, which is not always penalized due to the apparent good overall fit. In Chapter 3 this problem is discussed in more detail. Conceptual models are less prone to timing error effects and noisy output because of their fundamentally different model structure. However, the HBV results show that the model is not able to adequately simulate the recession limb of the hydrograph, which results in higher errors of the MSDE. High MSDE values for the ANN and HBV therefore seem to be related to different problems due to their specific model structures. The above proves that, while the MSDE gives extra information on model accuracy, it still does not allow for a completely effective numeric comparison of the ANN and the HBV models.

Figure 5.5 shows two hydrograph plots for the six-hour-ahead forecasts by both the ANN and the HBV model. The one-hour-ahead plot is disregarded since it is very similar to this figure, and its details have already been presented in Fig. 5.4. The gray area demarks the ranges of the forecasts made by the various Pareto solutions that were found, and the dots are the measurements. The big differences in the shape of these bounds reflect the fundamental differences between the two model approaches. The ANN bounds are wider because of the larger set of solutions, and they are skewed toward higher values because of some extreme solutions that prefer a lower MSDE over low-magnitude errors. The bounds of the HBV are more equally distributed around the measurements toward high and low values. The HBV bounds again show the model's inability to accurately simulate the shape of the recession curve. Note that the bounds mentioned here are not

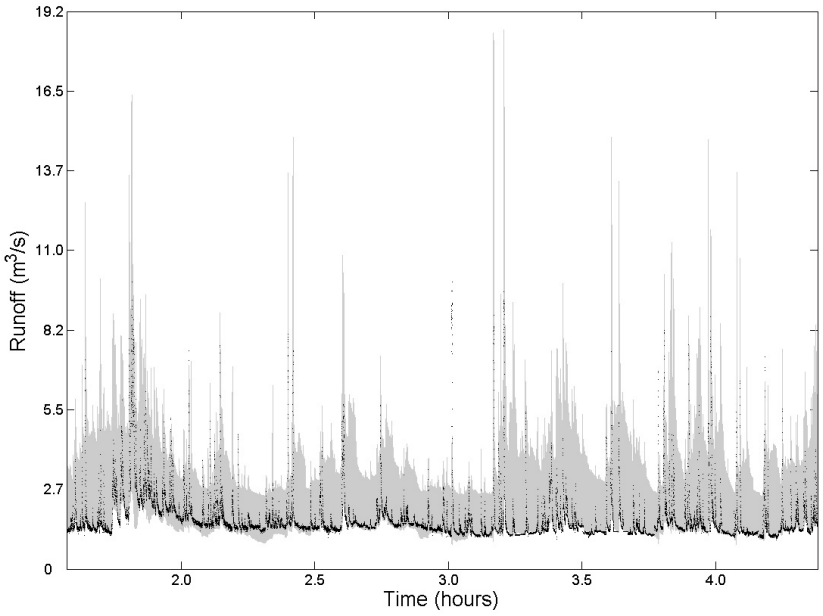
meant to be a formal representation of the models' uncertainty bounds. A more extensive training procedure (cf. the ANN versus the HBV calibration) results in higher apparent uncertainty estimates because of the larger spread of the Pareto solutions. This, however, does not necessarily imply a more realistic uncertainty assessment for the ANN model.

5.5 Summary and Discussion

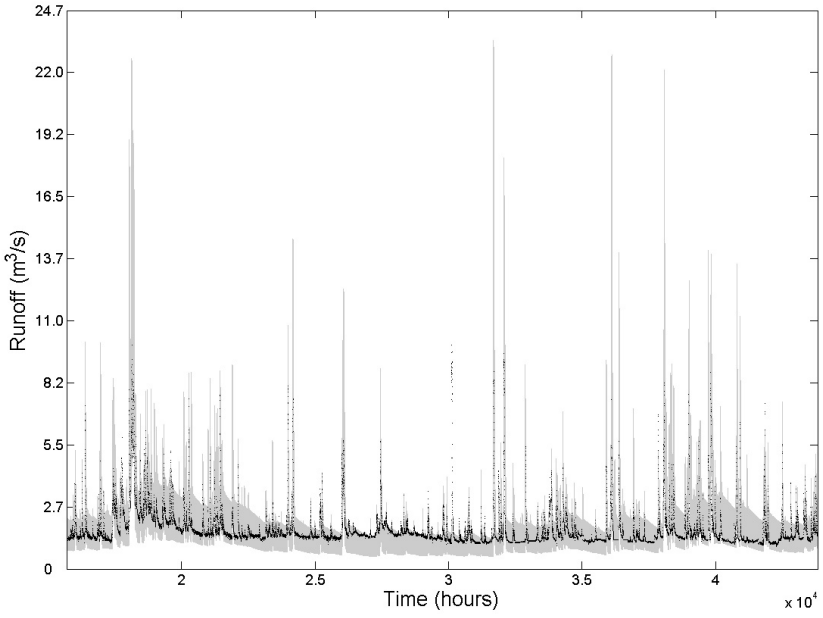
A multi-criteria comparison between the ANN and the HBV model using the NSGA–II optimization algorithm indicates that a single-criterion approach in both model calibration and evaluation is inadequate for evaluating and comparing these different model approaches. Using single objectives leads to disregarding information by drawing over-simplified conclusions on model performance and this prevents a deeper understanding of model approach and data. Moreover, by using the novel MSDE objective function, a more diverse set of solutions was found in comparison with only traditional objective functions, enabling a more extensive comparison of the models.

For one-hour-ahead forecasts, ANNs slightly outperformed the HBV model on the MSE and MSLE, but not on the MSDE. When the forecast horizon was increased to 6 hours, the HBV model outperformed the ANN model on all objective functions. However, the reasons for the differences in objective function values between the ANN and HBV models are different for each of the models. For example, the high MSDE for ANNs seems to be related to timing errors and noisiness of the simulation, whereas the HBV model has problems simulating the recession limb of the hydrograph. These conclusions show that better and perhaps more objective functions are still needed to allow a completely effective numeric evaluation of model performance.

The fundamental differences between the two model approaches tested here prevent straightforward and unequivocal comparisons from being made. The HBV model has physical laws, such as the mass balance equation, built into the model structure, whereas the ANN is a data-driven technique. Nevertheless, the implementation of the multi-criteria paradigm into both data-driven and conceptual modeling studies was shown to enable a more reliable and robust calibration and evaluation procedure, and to enable a more complete performance comparison of the two model approaches.



(a)



(b)

Figure 5.5: Variability in hydrograph simulations based on Pareto solutions of the six-hour-ahead forecasts af the (a) ANN model and (b) HBV model for the evaluation period. The dots represent the observed runoff values.

Chapter 6

Diagnostic Evaluation of Conceptual Rainfall–Runoff Models Using Temporal Clustering

Modified from:

de Vos, N. J., Rientjes, T. H. M., Gupta, H. V., 2009. Diagnostic evaluation of conceptual rainfall–runoff models using temporal clustering. *(submitted)*

Abstract

Given the structural shortcomings of conceptual R–R models and the common use of time-invariant model parameters, these parameters can be expected to represent broader aspects of the R–R relationship than merely the static catchment characteristics they are commonly supposed to quantify. In this chapter, the common assumption of time-invariance of parameters is relaxed, and instead signature information about the dynamics of model behavior and performance is sought. This is done by employing a temporal clustering approach to identify periods of hydrological similarity, allowing the model parameters to vary over the clusters found in this manner, and calibrating these parameters simultaneously. The diagnostic information inferred from these calibration results, based on the patterns in the parameter sets of the various clusters, is used to enhance the model structure. This approach shows how a diagnostic approach to model evaluation can be used to combine information from the data and the functioning of the hydrological model in a useful manner.

6.1 Introduction

The process of developing, calibrating and validating conceptual R–R models carries a significant degree of subjectivity. This subjectivity follows mainly from the various ways in which hydrological modelers can choose their preferred methods to evaluate model performance. Gupta *et al.* [2008] argue that the purpose of all evaluation must be diagnostic in focus, meaning that modelers need to identify those components of the model, which

when assumed to be functioning normally, will explain the discrepancy between model output and observed data. This can be approached through the identification of so-called signature information from the data. In the same paper, the authors discuss how model evaluation consists of three aspects:

1. Quantitative evaluation of model output, including statistical measures that express the difference between model output and observation time series.
2. Qualitative evaluation of model consistency, such as model sensitivity tests and visual inspections of model behavior.
3. Qualitative evaluation of model form and function, which implies a subjective expression of the degree in which the model structure and working comply with the real world system.

Quantitative model evaluation usually uses one or more statistical measures, in which commonly (a) the difference between a series of measurement data and a series of model outputs is transformed (e.g. power transformation, weighting), and then (b) the series of transformed differences is aggregated into a single value (e.g. by taking the mean squared error). These operations reflect a subjective choice by highlighting specific aspects of the hydrograph. Gupta *et al.* [1998] argued that using too few such aspects implies a loss of information since the calibration problem inherently involves many criteria. Following the development of effective and efficient algorithms (see Tang *et al.* [2006]), the power of the multi-criteria approach has been demonstrated in a number of hydrologic model calibration studies (e.g., Yapo *et al.* [1998]; Boyle *et al.* [2000]; Vrugt *et al.* [2003a]; Khu and Madsen [2005]; Kashif Gill *et al.* [2006]; Fenicia *et al.* [2007a]; de Vos and Rientjes [2007, 2008b]). However, in most multi-criteria approaches, when the residuals are aggregated into each statistic the information regarding model behavior and performance that is embedded in the time dimension is largely ignored. Arguably, it is questionable to ignore the time dimension given the predominantly dynamic nature of catchment runoff behavior.

Much research has been done to understand catchment runoff behavior and the mechanisms of runoff production (e.g. Betson and Marius [1969]; Hewlett and Hibbert [1963, 1967]; Dunne and Black [1970a, 1970b]; Freeze [1972a, 1972b]; Kirkby [1978, 1988]). That body of work indicates that the spatial domains over which runoff components are generated change over time. For example, in describing aspects that relate to runoff source areas and the saturation excess mechanism, Hewlett and Hibbert [1967] stress the importance of a belt of saturation lying along stream channels, that varies in width in response to rainfall, and forms a critical zone from which subsurface water and groundwater emerge to form the flood peak. These zones of saturation changes are largest during periods of extensive rainfall but are commonly not observable during periods of dryness. As such, meteorological forcings cause a catchment to have many response modes between high and dry weather flows, and cause the spatial domains from which runoff is produced to

change over time. In many storage-based conceptual models, the idea of changes in real world source areas is implicitly considered through the form of the storage-discharge relationship but changes in model parameter values to reflect dynamic changes that relate to such mechanisms are rarely (if ever) made. This example clearly shows how in R–R modeling complex dynamic relationships are commonly simplified and approximated by using time-invariant parameters.

Given the obvious structural shortcomings of conceptual R–R models and the use of time-invariant model parameters, it seems reasonable to proceed with a hypothesis that these parameters may represent broader aspects of the R–R relationship than merely the static catchment characteristics they are commonly supposed to quantify. By relaxing the common assumption of time-invariance of parameters, one can therefore attempt to obtain information from parameter variation about the dynamics of model behavior and performance. Several studies have reported on the calibration and evaluation of models with time-variant parameters, and on the subsequent extraction of information from the results. Wagener *et al.* [2003b] investigated the identifiability and evolution of model parameters over time for a very simple storage based runoff model using the DYNIA approach. Using a moving time window of fixed length over which parameter sensitivity and model performance were assessed, the approach suggested significant time variation of parameters and also revealed that such information could be used to develop insight into the model form and function. Choi and Beven [2007] used temporal clustering to identify periods of hydrological similarity. They subsequently evaluated predictions of Monte Carlo realizations of TOPMODEL parameter sets both within these periods and on multiple objective functions. The behavioral parameter sets were shown to vary significantly over both clusters and criteria. Moreover, no set was found that performed well on all clusters or on all criteria, indicating deficiencies in model structure. Another approach in which the time-invariance assumption of model parameters is relaxed, is to build models for specific parts of the hydrograph and find optimal ways to combine the results of the local models (e.g., Hsu *et al.* [2002]; Oudin *et al.* [2006]; Fenicia *et al.* [2007b]; Marshall *et al.* [2007]).

6.2 Goals and Scope

In this chapter an approach is developed and examined to diagnostic evaluation and improvement of a prior hydrological model structure by extracting temporal signature information via an augmented calibration procedure. The approach is based on the premise that deficiencies of the model structure cause the model parameters to vary with the hydrological modes of the system (if allowed to do so) to compensate for the effects of the model structural error. The main goals of this study are twofold: (1) to develop, test and

discuss a method for identifying signature information in the form of time-variant model parameter values for a R–R model of a meso-scale catchment and (2) to extract and use diagnostic information from the modeling results to improve the model structure.

To accomplish the first goal, a temporal clustering approach was devised to partition the historical data into several (here 12) periods of hydrological similarity. The model parameters were then permitted to vary (in a discrete manner) with time, taking on different values for each period of hydrological similarity, but remaining constant within each period; i.e. in this example each parameter can take on one of 12 different values over time, with the value corresponding to the temporal cluster mode active at that time. The goal of the approach is to see if the parameter variation can be related in some systematic manner to the magnitude of the system variables used to characterize periods of hydrological similarity, and to thereby make diagnostic inferences leading to improvements in the proposed hypothesis regarding the underlying structure of the system.

By applying the clustering procedure to observed data a physical basis is used in the dynamic analysis, effectively overcoming a main shortcoming of the previously mentioned approach by Wagener *et al.* [2003b] who used an arbitrary time window of fixed length. The clustering procedure proposed here has similarity to the approach used by Choi and Beven [2007], but extends it by the further logical step of actually interpreting the clustering results diagnostically so as to make improvements to the model. Note also, that the main goal is different from the approach mentioned above on combining local models. Although employing similar principles, the goal is to improve on a preconceived model structure rather than to identify and combine local model components.

The data set from the Leaf River catchment was used for this study (see Section A). Roughly a third of the data (October 1, 1948 to September 30, 1962) was used for calibration, and the rest for model evaluation. In the absence of observations regarding catchment storage, a synthetic time series of soil moisture was again generated using the simple soil moisture reservoir component of the GR4J lumped conceptual R-R model [Edijatno *et al.*, 1999; Perrin *et al.*, 2003].

6.3 Methods

6.3.1 Temporal Cluster Analysis

Introduction to Cluster Analysis

Cluster analysis is concerned with exploring data sets to assess whether or not they can be summarized meaningfully in terms of a relatively small number of clusters of objects which resemble each other and which are different in some respects from the objects in other clusters [A. K. Jain *et al.*, 1999; Everitt *et al.*, 2001]. The concept has been

applied in hydrology to cluster, for example, precipitation fields [Lauzon *et al.*, 2006], hydro-meteorological conditions [Toth, 2009], watershed conditions [Liong *et al.*, 2000], hydrological homogeneous regions [Frapporti *et al.*, 1993; Hall and Minns, 1999], and also in regionalization approaches [Burn, 1989; Srinivas *et al.*, 2008]. In this work, an attempt is made to find periods of hydrological similarity by temporal clustering of hydrological data. By this approach, the information contained in the dimension of time is compressed into a discrete set of clusters and its information can be meaningfully and conveniently summarized.

Cluster Inputs

Four time series were chosen for the first cluster analysis: (1) the ten-day moving average of the precipitation (P_{ma}), (2) the GR4J-simulated soil moisture (S), (3) the natural logarithm of the discharge (Q_{ln}), and (4) the first derivative of the discharge (dQ). These variables represent information regarding the recent input, memory, output and dynamics of the catchment, respectively. The logarithmic transformation of discharge is performed to reduce the skewness of the discharge distribution. The input to the clustering algorithm consists of the simultaneous variable values at each time t, so the algorithm had four inputs. A second cluster analysis was also performed using (1) precipitation (P), (2) the ten-day moving average of the precipitation (P_{ma}), and (3) the GR4J-simulated soil moisture (S). Since this second analysis does not rely on knowledge of the output of the catchment, it can be used in prediction mode.

Clustering Algorithm

The k-means clustering algorithm involves calculation of the centroid of a fixed number of clusters. This is usually done as proposed by Lloyd [1982]. The method can be summarized as follows.

1. Randomly choose k initial centroids $C = \{c_1, \ldots, c_k\}$.
2. Set each cluster N_i to be the points in X that are closer to c_i than to any other centroid.
3. Set each c_i to be the centroid of all points in N_i.
4. Repeat steps 2 and 3 until C is stable.

The proximity measure used to determine the closeness of points to centroids was the Euclidian distance. The k-means++ seeding method [Arthur and Vassilvitskii, 2007] was used to choose the initial centroids with a probability proportional to the density of points. Arthur and Vassilvitskii [2007] show that this approach can significantly reduce errors and improve convergence speed of the algorithm.

The *a priori* choice of the number of clusters was made by running the clustering procedure for 2 to 30 clusters using fuzzy clustering and studying the partition coefficient and the

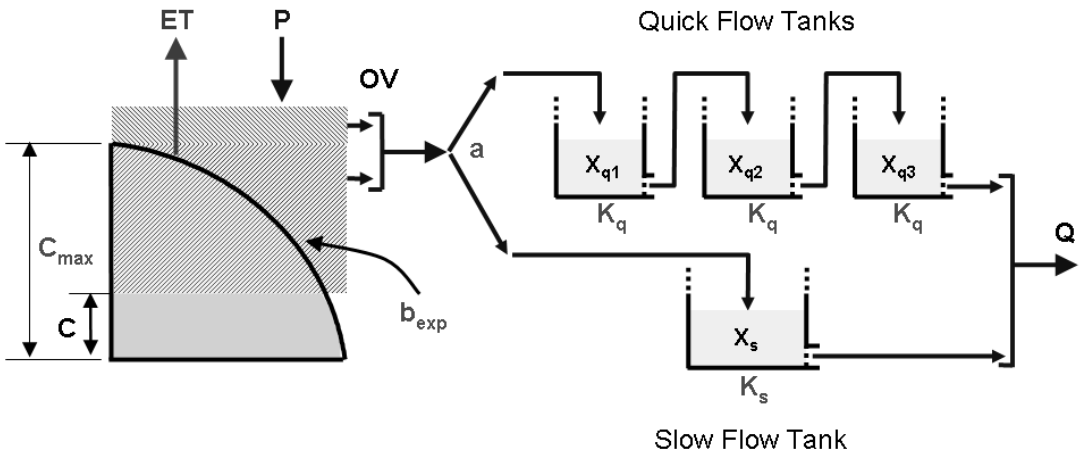


Figure 6.1: The HyMod model structure.

Xie and Beni [1991] index for each of these runs. These validity measures are commonly used for expressing the quality of a fuzzy clustering procedure. The number of clusters was set to 12 based on the fact that both measures did not significantly improve for a larger number of clusters. In comparison, Choi and Beven [2007] found 15 clusters to be appropriate for their data set.

6.3.2 Conceptual Rainfall-Runoff Model

The five-parameter conceptual HyMod R–R model, shown in Figure 6.1, was used for this study. This model is based on the probability distribution model by Moore [1985] and was introduced by Boyle [2000]. It was applied more recently by Wagener *et al.* [2001] and [Vrugt *et al.*, 2003b] among others. HyMod consists of a simple two-parameter rainfall excess model, in which it is assumed that the soil moisture storage capacity C varies across the catchment and, therefore, that the proportion of the catchment with saturated soils varies over time. The spatial variability of C is described by the following distribution function:

$$F(C) = 1 - \left(\frac{1 - C(t)}{C_{max}}\right)^{b_{exp}} \tag{6.1}$$

where C is always smaller than C_{max}. The routing component consists of a series of three linear reservoirs for quick flow and one linear reservoir for slow flow. Table 6.1 describes the HyMod parameters and presents reasonable ranges to be used in constraining their calibration (cf. Vrugt *et al.* [2003b]). The additional parameters that are mentioned in this table will be explained later in this chapter.

Table 6.1: HyMod model parameters.

Name	Description and unit	Prior range
C_{max}	Maximum soil moisture content [L]	10–1500
b_{exp}	Spatial variability of soil moisture capacity (-)	0.01–1.99
A	Quick/slow flow distribution factor (-)	0.01–0.99
K_s	Residence time slow reservoir (T)	0.01–0.99
K_q	Residence time quick reservoirs (T)	0.01–0.99
	Additional parameters:	
F_{RFC}	Rainfall correction factor (-)	0.5–1.5
L_s	Length transformation function slow reservoir (T)	1–20
L_q	Length transformation function quick reservoir (T)	1–40

6.3.3 Model Calibration

Traditional and Dynamic Calibration Approaches

The HyMod model is calibrated twice, once using a traditional calibration approach with time-invariant parameters and once using a dynamic calibration approach with time-variant parameters. In the latter procedure, the model parameters are allowed to take on different values for each of the 12 different clusters, resulting in a number of degrees of freedom equal to the number of parameters times 12. For both calibration approaches, a single objective function was used to express the error over the entire calibration time series. The dynamic calibration approach results in 12 parameter sets, each of which optimizes model performance for the specific temporal cluster that the system is in. This dynamic calibration is a rather brute force approach and the number of parameters that is calibrated seems to conflict with principles of parsimony. Note, however, that the goal of this calibration approach is to investigate the temporal variability of parameters, rather than to find the best-performing model per se.

For both calibration approaches, the Normalized Root Mean Squared Error (NRMSE) are used as objective function:

$$NRMSE = \frac{\sqrt{\frac{1}{K}\sum_{k=1}^{K}\left(\hat{Q}_k - Q_k\right)^2}}{\frac{1}{K}\sum_{k=1}^{K} Q_k} \tag{6.2}$$

where K is the total number of data elements, and $\hat{Q}_k$ and Q_k are the simulated and the observed discharges at the kth time interval respectively. Because the NRMSE is a normalized error statistic, the performance on clusters of different lengths can be compared.

Optimization Algorithm

Parameter optimization was performed using a self-adaptive variant of the Differential Evolution (DE) algorithm introduced by Storn and Price [1997]. While relatively simple, the algorithm is powerful, and generally shows high accuracy and fast convergence on many test problems (see Vesterstrøm and Thomsen [2004]). It has been applied to hydrological model calibration by Shoemaker *et al.* [2007].

Several variants of the DE algorithm have been suggested. Here, the commonly used $DE/rand/1/bin$ strategy is selected, which can be summarized as follows. A population of N individuals $\mathbf{x_{i,G}}$, $i = 1, 2, \ldots, N$, each of which is a vector of D optimization parameters, is evolved for a number of generations (indicated by G). The evolution is defined as a process of three operations: mutation, crossover, and selection. Each individual $\mathbf{x_{i,G}}$ is mutated according to

$$\mathbf{v}_{i,G+1} = \mathbf{x}_{r_1,G} + F \cdot (\mathbf{x}_{r_2,G} - \mathbf{x}_{r_3,G}), \; r_1 \neq r_2 \neq r_3 \neq i \tag{6.3}$$

with randomly chosen indices $r_1, r_2, r_3 \in [1, N]$. $F \in [0, 2]$ controls the amplification of the difference vector $(\mathbf{x}_{r_2,G} - \mathbf{x}_{r_3,G})$ and is one of the two main control parameters of the algorithm. If any component of a mutant vector falls outside the acceptable parameter bounds, it is set to the bound value. Crossover is performed using the individuals and their mutants according to

$$\mathbf{u}_{i,G+1} = (u_{1i,G+1}, u_{2i,G+1}, \ldots, u_{Di,G+1}) \tag{6.4}$$

where

$$u_{ji,G+1} = \begin{cases} v_{ji,G+1}, & \text{if } r(j) \leq C \text{ or } j = r_n(i) \\ u_{ji,G}, & \text{if } r(j) > C \text{ and } j \neq r_n(i) \end{cases} \tag{6.5}$$

for $j = 1, 2, \ldots, D$. $r(j) \in [0, 1]$ is the jth output of a uniform random number generator. $C \in [0, 1]$ is the crossover constant and is the second main control parameter of the DE algorithm. $r_n(i) \in (1, 2, \ldots, D)$ is a randomly chosen index which ensures that at least one element of $\mathbf{u}_{i,G+1}$ comes from $\mathbf{v}_{i,G+1}$. Selection is performed according to a greedy selection scheme:

$$\mathbf{x}_{i,G+1} = \begin{cases} \mathbf{u}_{i,G+1}, & \text{if } f(\mathbf{u}_{i,G+1}) \text{ is better than } f(\mathbf{x}_{i,G}) \\ \mathbf{x}_{i,G}, & \text{otherwise} \end{cases} \tag{6.6}$$

for $j = 1, 2, \ldots, D$. This way, the old individual is replaced only if the objective function value of the new individual is better.

A crucial issue for the efficiency and efficacy of the DE algorithm is the choice of values for its control parameters F and C [J. Liu and Lampinen, 2002; Brest *et al.*, 2006]. Here the approach suggested by Brest *et al.* [2006] is followed, in which the control parameters are evolved by placing them inside the vector associated with each individual. This self-adaptive version has been shown to successfully find optimal control parameters for different problems and consequently outperform other implementations of DE.

The population size for the traditional calibration procedure was set to 10 times the dimension of the optimization problem (i.e., the numbers of parameters to be optimized). The number of generations was limited to 250. For the dynamic calibration, the population size was 5 times the dimension of the problem (i.e., the numbers of parameters to be optimized times the number of clusters) and $1,000$ generations. To initialize the second calibration procedure the initial parameter values were set close to the optimum found during the traditional calibration, thereby speeding convergence.

6.4 Results and Analysis

6.4.1 Cluster Analysis

The clustering results in Figure 6.2 show the four-dimensional simulation data classified into 12 clusters. The clusters have been numbered according to their rank in terms of magnitude of soil moisture for convenience in presentation of subsequent results. The figure shows that an increase of P_{ma} (moving average of precipitation) generally results in an increase of S (soil moisture), as is to be expected. Although there is likely to be some overlap in the information content of these variables, the clarity (minimal overlap) of the clusters in this plot indicates that the clustering algorithm has selected these two dimensions as major variables for distinguishing between clusters. In a similar manner, the obvious relationship between S and Q_{ln} (log discharge) has also been exploited by the clustering algorithm. In contrast, the relationships between Q_{ln} and dQ (discharge derivative) and between P_{ma} and dQ, which more strongly reflect the high flow dynamics of individual events, and the dimension of dQ, do not seem to have been strongly used for cluster discrimination.

Similar results are observed for the three-dimensional prediction data (Figure 6.3), where S and P_{ma} again are the primary factors determining the shape and location of the clusters. Here, in contrast with the simulation data, information on streamflow (Q_{ln} and dQ) is missing and so the clustering algorithm appears to be using P (precipitation) more actively as an indicator for distinguishing peak flow events (cluster 9, for example, is poorly discernable in the S versus P_{ma} subplot but very pronounced in the two other subplots).

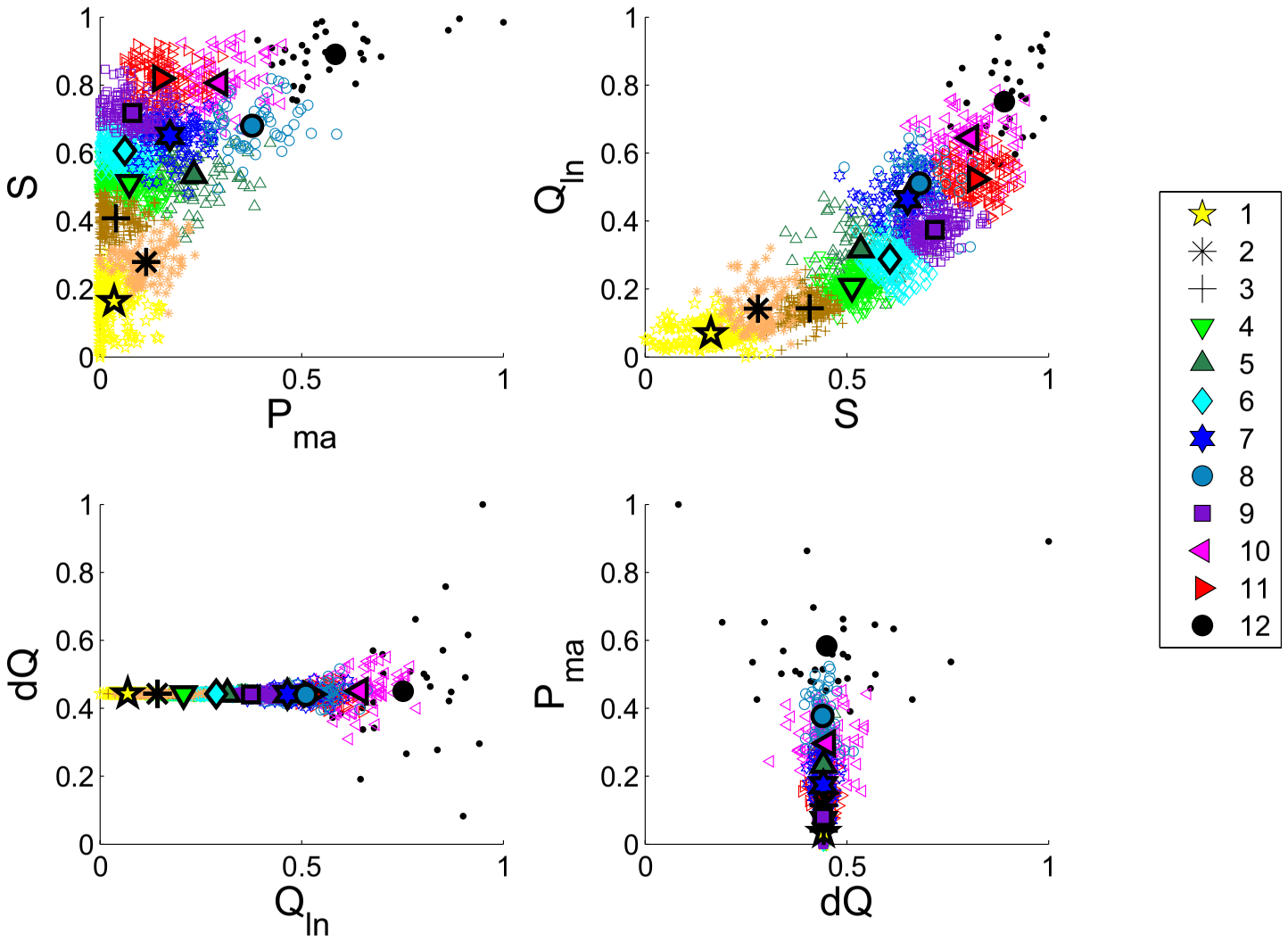


Figure 6.2: Two-dimensional projections of clustering results on the four-dimensional simulation data set. Data are normalized between 0 and 1. The cluster centroids are indicated with special markers. The clusters are sorted according to their rank in soil moisture for convenience (see legend).

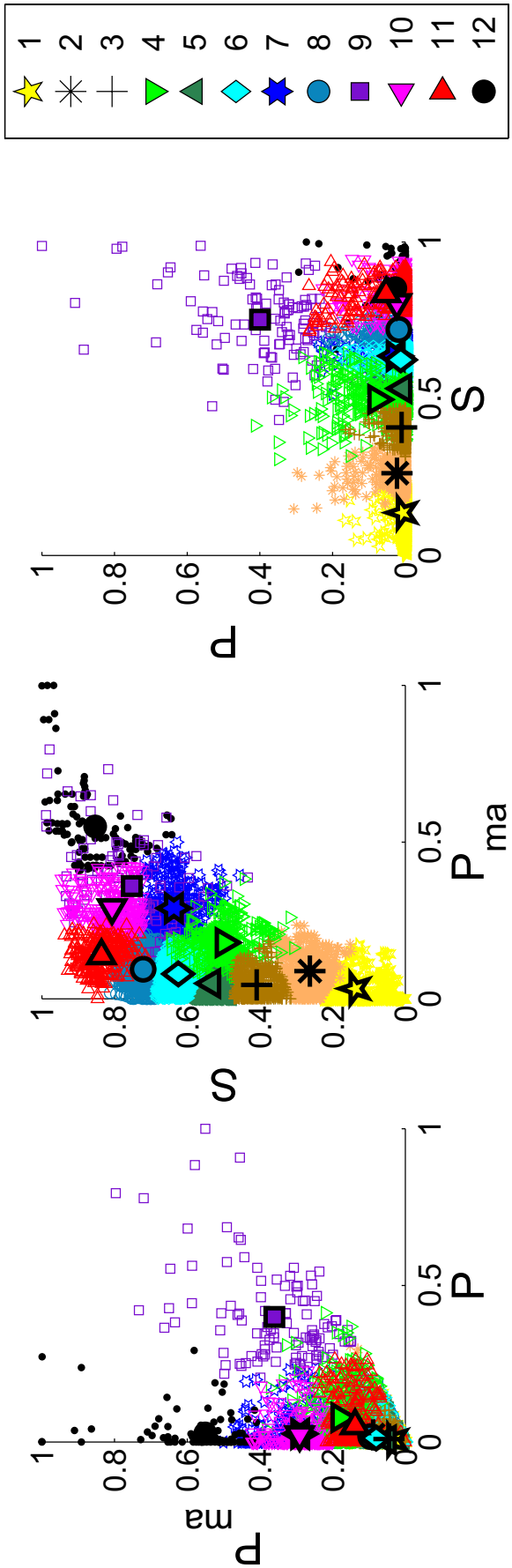


Figure 6.3: Two-dimensional projections of clustering results on three-dimensional prediction data set. Data are normalized between 0 and 1. The cluster centroids are indicated with special markers. The clusters are sorted according to their rank in soil moisture for convenience (see legend in Figure 6.2).

Figure 6.4 shows the temporal clusters identified from the prediction data (not using streamflow information) plotted along with the hydrograph for the evaluation period. Given that the hydrograph information was not in the clustering procedure, and the fact that these are evaluation period results, the close concurrence of the patterns indicates that the temporal clustering has been quite successful. The periods of wetting and drying, and the related runoff dynamics of the catchment are clearly reflected in the organization of the clusters on both the event and seasonal scales.

After having analyzed the clustering procedure with both simulation and prediction data, only the prediction clustering data is presented in the remainder of the chapter for reasons of brevity. Consequently, the clustering and model setup uses no information from current discharges and hence reflects a more challenging prediction scenario.

6.4.2 Model Performance

The HyMod model calibration and evaluation performance statistics for the two calibration approaches are presented in Table 6.2. Each row shows the NRMSE performance for the portion of the data corresponding to one of the prediction clusters when either the traditional or dynamic calibration strategy is employed. In interpreting these results, please note that the larger cluster numbers indicate larger amounts of soil moisture storage (catchment wetness; see Figures 6.2 and 6.3). The last row shows the same statistic computed for the entire period. Despite having more degrees of freedom, and producing smaller overall NRMSE (better overall model performance), the dynamic calibration has resulted in worse evaluation period performance on clusters 2 to 8 which correspond to drier periods. It appears that the dynamic calibration has found a solution that provides better predictions for the high flows than on low flows, which are severely underestimated. Moreover, the total discharge volume of the calibrated model was about 10% less than the observed volume. The above indicates a shortage of water, especially in the slow flow reservoir, and it is therefore assumed that the model receives too little net inflow by meteorological forcing. It appears that, in trying to minimize the overall period performance statistic for the current model structure, the calibration process consequently has settled on a trade off which emphasizes the fit of high flows over low flows.

To try and improve model performance, therefore, a rainfall correction factor (an additional parameter FRFC; see Table 6.1) was first added to the model to compensate for measurement errors through multiplication with the rainfall (Equation 6.7); the enhanced version of the model is given the name "HyMod–N1". The use of a correction factor is not uncommon in hydrological models and is used in, for example, the HBV model [Lindström *et al.*, 1997]. Recalibration using the traditional calibration method resulted in an optimal value of $P_{corr} = 1.057$, indicating that the rainfall data likely underestimate the actual rainfall by almost 6%. Optimal values (using traditional calibration) for all parameters of

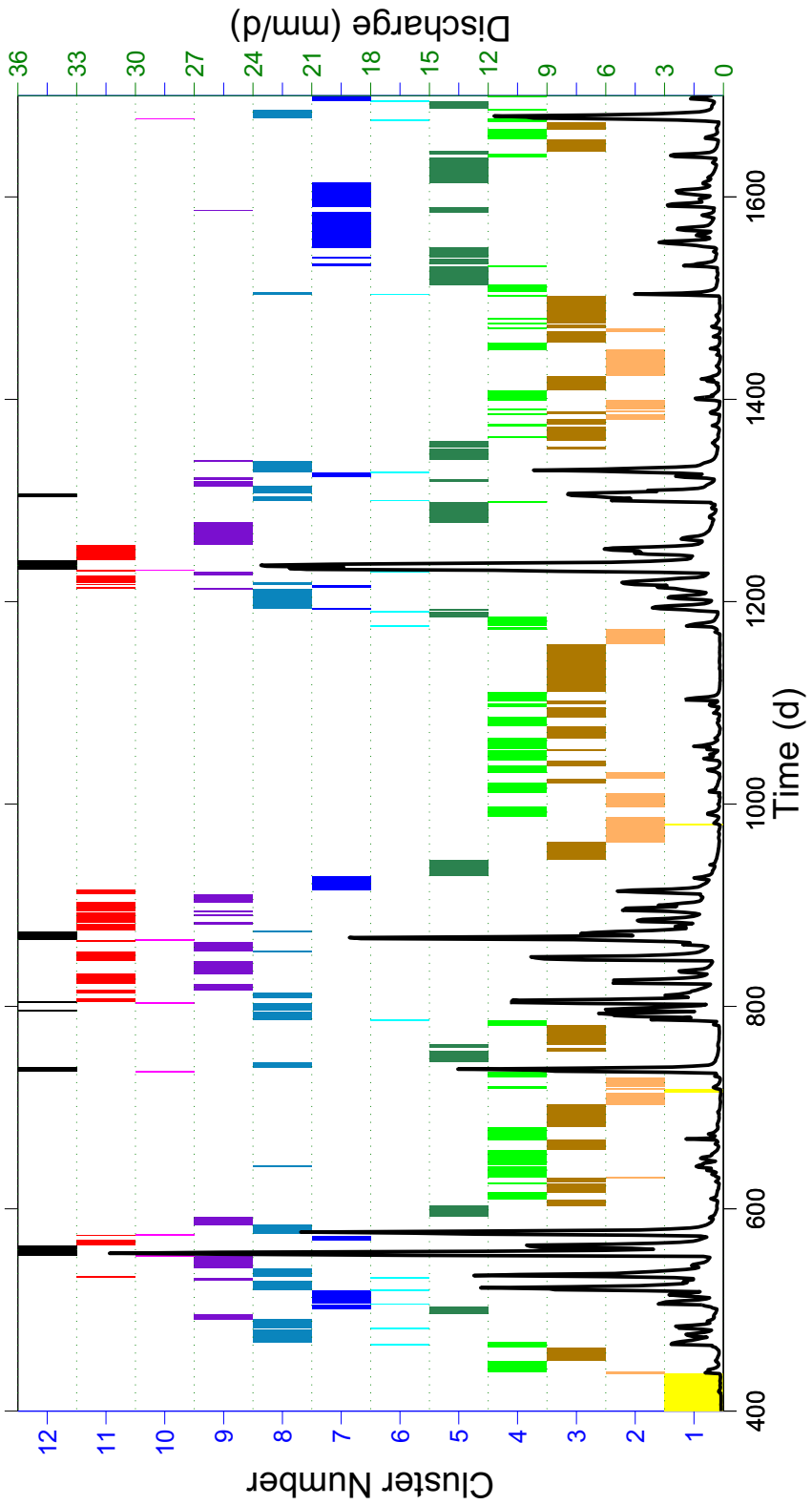


Figure 6.4: Clustering results for the prediction data set. This is a representative detail from the evaluation period, showing the clusters in time, along with the observed hydrograph.

Table 6.2: HyMod performance expressed over the clusters plus overall performances for calibration and evaluation periods using dynamic (based on simulation clustering) and traditional calibration procedures. Highlights indicate that the traditional or dynamic calibration approach outperforms the other.

	Calibration NRMSE		Evaluation NRMSE	
Cluster no.	Traditional calibration	Dynamic calibration	Traditional calibration	Dynamic calibration
1	0.945	0.672	0.943	0.742
2	0.883	0.890	0.796	0.823
3	0.866	0.884	0.826	0.832
4	0.704	0.646	0.687	0.693
5	0.709	0.831	0.741	0.793
6	0.631	0.767	0.625	0.766
7	0.465	0.455	0.565	0.568
8	0.508	0.532	0.532	0.547
9	0.477	0.449	0.523	0.455
10	0.433	0.343	0.374	0.377
11	0.408	0.367	0.455	0.383
12	0.339	0.274	0.338	0.330
Total	0.836	0.702	0.793	0.773

Table 6.3: Optimal parameter values found using traditional calibration for all HyMod model variants.

Parameter	HyMod	HyMod–N1	HyMod–N2
C_{max}	210	241	248
b_{exp}	0.400	0.425	0.440
A	0.296	0.270	0.214
K_s	0.319	0.295	0.350
K_q	0.823	0.830	0.927
F_{RFC}	-	1.057	1.044
L_s	-	-	2
L_q	-	-	2

the original HyMod model and the enhanced HyMod–N1 model are presented in Table 6.3.

$$P_{corr} = F_{RFC} \cdot P \tag{6.7}$$

The five original parameters of the HyMod–N1 model were subsequently dynamically re-calibrated in the same way as for the HyMod model, but with the additional F_{RFC} parameter kept constant. The assumption here is that the rainfall underestimation is of a more structural nature, and the dynamic calibration should not abuse the potentially influential F_{RFC} parameter to compensate for other errors. The results presented in Table 6.4 clearly show that the trade-off between high and low flow performance has decreased. The smaller calibration and evaluation period errors achieved by the dynamic calibration, both overall and for each cluster, indicate that the rainfall correction factor has fulfilled its purpose and is now allowing the dynamic character of the calibration process to be better exploited.

In the next step of the diagnostic approach the information about functioning of model structure that is implicitly contained in the variability of the parameters over the 12 clusters was made use of. The subplots in Figure 6.5 show the optimal values for each parameter for each temporal cluster, found using the dynamic calibration procedure; the horizontal line represents the time-invariant parameter value found using traditional calibration. Of course, many patterns might be hidden in these results, and useful information could be difficult to detect due to complex parameter interactions. Here a start is made with the simple test of hypothesis that useful information can be extracted from each individual dynamic parameter set and from their most obvious coincident patterns of variation. Two patterns stand out clearly in Figure 6.5: the general tendency for the K_s and K_q recession coefficients to be smaller than their "normal" (time-invariant) value, and the tendency for both of these parameters to increase with increasing wetness.

Table 6.4: HyMod–N1 performance expressed over the clusters plus overall performances for calibration and evaluation periods using dynamic (based on simulation clustering) and traditional calibration procedures. Highlights indicate that the dynamic calibration outperforms the traditional calibration.

Cluster no.	Calibration NRMSE		Evaluation NRMSE	
	Traditional calibration	Dynamic calibration	Traditional calibration	Dynamic calibration
1	0.946	0.658	0.932	0.665
2	0.881	0.656	0.773	0.696
3	0.799	0.808	0.776	0.765
4	0.794	0.607	0.756	0.635
5	0.755	0.723	0.788	0.685
6	0.646	0.529	0.627	0.502
7	0.437	0.454	0.537	0.554
8	0.501	0.402	0.530	0.425
9	0.509	0.484	0.525	0.428
10	0.418	0.335	0.366	0.368
11	0.403	0.338	0.471	0.372
12	0.324	0.262	0.327	0.333
Total	0.810	0.670	0.779	0.749

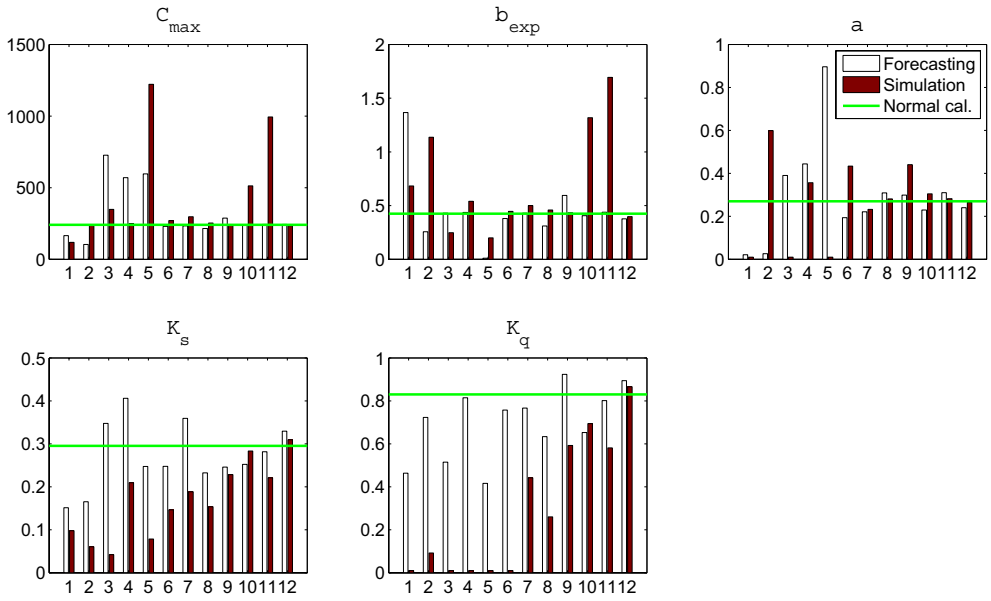


Figure 6.5: Estimates of the optimal values of the 5 dynamic HyMod–N1 parameters for the 12 clusters. The lines show the optimal values found by the traditional calibration procedure.

These patterns suggest that the slow and the quick flow reservoirs may be functioning sub-optimally, and might be better represented using nonlinear recession rate dynamics. However a test of this hypothesis did not lead to significant performance improvements. Returning to Figure 6.5 it is noted that smaller values of the recession coefficients are preferred during dry periods, but during wetter periods the recession coefficients are close to the "normal" values. This observation provides the hint that the model is attempting to restrict the reservoir storage outflow both at the beginning and at the end of each storm event. This kind of behavior can be achieved by incorporating a triangular transformation function before the flow reservoirs that results in a more gradual input to the reservoirs (a moderating of the instantaneous rainfall impulse shocks); a similar modification was proposed by Fenicia *et al.* [2008] to improve the structure of another hydrological model.

Implementation of this concept into HyMod–N1 results in the updated model structure HyMod–N2. Two additional parameters, L_s and L_q, were introduced which represent the base lengths of the triangular functions (see Table 6.1), and the weights that compose the transformation function are calculated according to:

$$b_i = \frac{4i/L^2}{B} \qquad \text{for} \qquad i \in 1, 2, ..., L \tag{6.8}$$

where

$$B = \sum_{i=1}^{L} b_i \tag{6.9}$$

When the HyMod–N2 model is recalibrated using the traditional (time-invariant parameter) calibration procedure the result is the overall and individual cluster performance results shown in Table 6.5. Note that no dynamical calibration is used here, because the test is whether the improved model structure has reduced the need for the parameters to vary in time. The optimal (time-invariant) parameter values are again shown in Table 6.3 for easy comparison with the values obtained for the previous model structures. The results show that HyMod–N2 generally provides better performance than HyMod–N1 model on both the calibration and evaluation data. This is not the case for several dry clusters, but this may be due to the use of an objective function that focuses more on high flows. Interestingly, a comparison of Tables 6.4 and 6.5 shows that the HyMod–N2 (with time-invariant parameters) performs even better than the dynamically calibrated HyMod–N1 model on the evaluation data, supporting the hypothesis that the added transformation functions indeed results in a more consistent model of the R–R process.

Figure 6.6 provides another performance comparison of the three model structure variants, this time after calibration with the MOSCEM–UA multi-criteria algorithm [Vrugt *et al.*, 2003b]. The settings of the algorithm that were used are as follows: number of complexes equal to the number of parameters to be calibrated, 100 random samples per complex

Table 6.5: HyMod-N2 performance, expressed by clusters plus overall performances, for calibration and evaluation periods using traditional calibration procedure. Highlights indicate that the HyMod-N2 model outperforms the HyMod-N1 model.

Cluster no.	Calibration NRMSE	Evaluation NRMSE
1	0.952	0.938
2	0.877	0.775
3	0.811	0.788
4	0.767	0.728
5	0.779	0.785
6	0.656	0.635
7	0.413	0.499
8	0.498	0.518
9	0.453	0.478
10	0.410	0.352
11	0.384	0.431
12	0.293	0.291
Total	0.760	0.723

and total number of draws equal to 2,000 times the number of complexes. The two objective functions used in the calibration are the Mean Squared Error (MSE) and the MSE of the log-transformed discharges (MSLE), which emphasize errors on high flows and low flows, respectively. The Pareto plot with the evaluation period results of the model structures show improvements in the successive model iterations for both the MSE and the logMSE. The improvements in the latter show that the HyMod–N2 is indeed capable of outperforming the HyMod–N1 model on low flows, suggesting that the results in Tables 6.4 and 6.5 are not fully representative of model performance because of the use of a single objective function which focuses on high flows. Although a clear trade-off between the two objective functions remains, the smaller spread of the solutions of the HyMod–N2 model compared to the other models indicates that this model has a more robust model structure. Finally, the parameter variability of the solutions found by the MOSCEM–UA algorithm (not shown here) is slightly reduced with each successive model structural iteration.

6.5 Summary and Discussion

This chapter has made a first step towards examining and understanding how to conduct diagnostic model evaluation by extracting temporal signature information via an

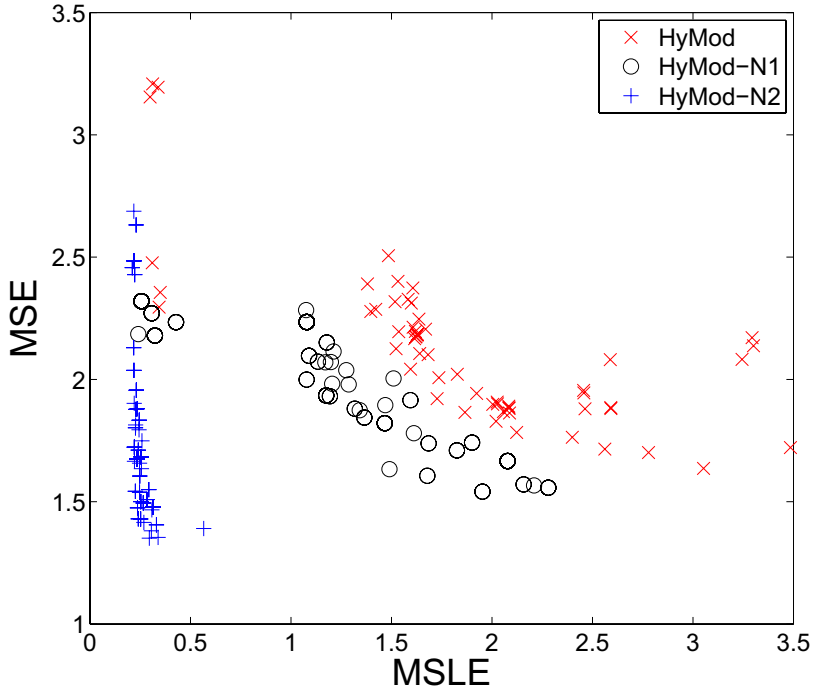


Figure 6.6: Multi-criteria evaluation results for the three model structures after calibration with the MOSCEM–UA algorithm.

augmented calibration procedure based on temporal clustering. Such an approach might be used to recursively achieve diagnostic improvements of hydrological models. The results show that consistent patterns of parameter variation do indeed show up through the application of this approach, and that analysis of these patterns can point towards potential model improvements. In this work two iterations are explored resulting in updates to the model structure, based on quite obvious patterns. More powerful alternatives to this simple diagnostic approach are likely to exist. Nevertheless, the research presented here illustrates one pathway to the development of a diagnostic approach to model evaluation. The study example shows how information from the data and the functioning of the hydrological model can be combined in a useful manner to achieve improvements to the working model hypothesis.

What can be concluded from this work is that, although process knowledge and perceptual models of reality will probably remain the most important source of information for model development and improvement, there is still much that can be gained through careful scrutiny of data and model functioning. The issue to be better understood is how such a diagnostic evaluation should be conducted — what are the strategies that will generally lead to uncovering information that can be reliably and readily used for model improvement. For this, a variety of model diagnostic approaches will need to be tested and verified. Further research on the challenging step in which the calibration information

is translated into model structural improvements could be especially beneficial. More advanced data analysis or pattern recognition techniques could prove to be particularly useful in this.

Other conclusions and recommendations are:

- The k-means clustering algorithm was shown to be effective in identifying hydrologically similar periods. Future research might benefit from the use of more sophisticated clustering techniques such as fuzzy clustering or random forests [Breiman, 2001]. The rather subjective choices in the clustering procedure need to be further investigated to find appropriate settings for hydrological applications.
- The self-adaptive DE optimization algorithm was found to be effective in hydrologic model calibration. The algorithm obtained good solutions, within a reasonable number of function evaluations, for both the traditional calibration and on the more complex (higher dimensional) dynamic calibration.
- Although the self-adaptive DE algorithm is powerful, it does not provide estimates of the parameter uncertainty of its results. A method that helps to assess the uncertainty in the parameter estimates would significantly benefit the diagnostic model evaluation.

Chapter 7

Conclusions and Recommendations

7.1 On Computational Intelligence in Rainfall–Runoff Modeling

The previous four chapters have shown investigations on the application of CI techniques in R–R modeling, and have indicated several specific advantages and disadvantages of such techniques in the development and evaluation of R–R models. Although the assumptions underlying this work and the shortcomings of the methods used prevent broad generalizations, this section will attempt to summarize and synthesize these findings with respect to the frameworks mentioned in Chapter 1. The following three subsections present the conclusions of the main fields of application, and show in what way this work has advanced the field of hydrological modeling.

7.1.1 Artificial Neural Networks as Data-Driven Models

- This work is among the first to call to attention timing errors issues in ANN R–R models. Some of the ANN R–R models in Chapter 3 of this work suffer from timing errors, which is shown to be the result of a dominating autoregressive component. This component is introduced by using highly autocorrelated previously observed runoff values as ANN model input. As a result, the ANN model in fact does little more than presenting the model input as output. Most commonly-used objective functions do not penalize such timing errors, which is why they often go unnoticed and why they pervade the literature on ANN R–R modeling.
- This work is also among the first to investigate solutions to the timing error issue (Chapter 3). Using a combination of alternative input variables for representing the hydrological state of a catchment (for example, moving averages of rainfall time series or soil moisture time series) and a proper objective function that penalizes the model for having timing errors, the issue is shown to be somewhat alleviated.
- Chapter 5 presents the novel approach of comparing an ANN and a conceptual R–R model using multiple criteria. The ANN model performed slightly better than the

conceptual model for short-term forecasting, but was outperformed with increasing forecast lead times.

7.1.2 Computationally Intelligent Parameter Estimation

- In this work, various results are presented that support the increasingly accepted idea in hydrological modeling that using multiple criteria in the calibration and evaluation of models is necessary to extract more information contained in data and to better evaluate models.
  - In Chapter 3, it is shown that not all differences between modeled and observed hydrograph characteristics can be adequately expressed by a single performance criterion since the results indicated that there seemed to be a trade-off between the objectives of correct timing and good overall fit for an ANN R–R model.
  - Chapter 4 subsequently presents one of the first applications of MC optimization algorithms to ANN R–R models. Similar to MC calibration of conceptual hydrological models, trade-offs between objective functions also manifest themselves in MC ANN training. The results indicated that by using MC training, more stable regions in the weight space can be identified which result in more reliable models compared to SC training.
  - The case for MC calibration of both data-driven and conceptual hydrological models in again made in Chapter 5, through one of the first MC comparisons of such models. It was again shown that the use of single objectives leads to disregarding information by drawing oversimplified conclusions on model performance, preventing a deeper understanding of model and data.
- By comparing many different optimization algorithms for the training of ANN R–R models, this work shows that the sensitivity of ANN model results on the method of optimization are larger than usually expected by hydrological modelers. Whether it be in differences between various local algorithms (Chapter 3), between local and global algorithms, or between SC and MC algorithms (both in Chapter 4), the choice of algorithm has large effects on model accuracy and uncertainty. Chapter 4 shows that there are also large differences in *a posteriori* weight values of the ANN models after training using different algorithms.
- A new objective function was proposed in Chapter 4 of this work, which penalizes a model for errors regarding hydrograph timing and shape by looking at differences between the first derivatives of the simulated and observed times series. This Mean Squared Derivative Error clearly complements traditional objective functions and therefore helps in extracting more information from the data.
- The self-adaptive DE optimization algorithm was found to be effective in one of its first applications in hydrologic model calibration (Chapter 6). The algorithm

obtained good solutions within a reasonable number of function evaluations, for both a low-dimensional and high-dimensional optimization problem.

7.1.3 Hydrological Clustering

- A temporal clustering approach is employed in Chapter 6 to identify periods of hydrological similarity. The model parameters were subsequently allowed to vary over the clusters found in this manner, and these parameters were calibrated simultaneously. The parameter variability between the hydrologically similar periods was subsequently used to make diagnostic inferences leading to improvements in the proposed model structures. This diagnostic step represents a successful novel application of clustering techniques that addresses the challenging and fundamental hydrological issue of how to achieve improvements on the working model hypothesis, and as such hints at new possibilities in hydrological modeling.
- The k-means clustering algorithm was shown to be effective in identifying hydrologically similar periods from a data set.

7.1.4 Synthesis

After using CI for (1) system identification, (2) parameter estimation, and (3) data mining in R–R modeling, the overall conclusion is that applications of CI in R–R modeling show a lot of promise. This supports the findings in recent literature, as presented in Chapter 2. CI methods can be considered powerful thanks to, for example, their general effectiveness in dealing with nonlinearity (e.g., ANNs as R–R models) and high-dimensionality (e.g., effectiveness of CI parameter estimation). Moreover, CI methods provide an alternative viewpoint on hydrological data and models that is strikingly different from traditional methods, and as such are able to extract information hitherto overlooked.

Nevertheless, there are some pitfalls of CI methods. For example, their application to system identification in R–R modeling runs the risk of producing a model that is not well-performing, robust or consistent if it is done without the use of process knowledge in model development or evaluation (a clear example of this are the often-overlooked timing errors of ANN R–R models). Better guidelines on data-driven model development and calibration in R–R modeling would help, but most CI methods remain too flexible methods to be considered realistic approximations of the natural system (data-driven models commonly disregard the water balance, for example). The advantages of an approach in which models are built on physical foundation but make use of sophisticated algorithms that exploit the data better, are therefore obvious: the complementary characteristics of knowledge-driven and data-driven modeling can then be optimally exploited. Recent research, including this work, shows that by carefully combining both process knowledge, modeling experience

and intuition (see [Savenije, 2009]), it is possible to forge new model development and evaluation methods that combine the best of both worlds.

Regarding the calibration of hydrological models, CI parameter estimation techniques seem to generally outperform traditional methods. Compared to traditional methods, however, CI methods often have increased computational demands. This work has also shown the desirability of MC calibration of models and how CI methods are effective in this respect.

No formal quantification of uncertainty of model output has been performed in this work. However, the issues related to parameter uncertainty discussed in Chapters 4 and 5 translate qualitatively to the uncertainty of model output. Also, MC calibration can reduce parameter uncertainty because of the increased amount of information that is extracted from model and data. Moreover, the findings of Chapter 6 strongly suggest the possibility of finding model structural improvements, which also leads to reduced model structural uncertainty in R–R modeling.

This work has shown that even without the use of additional information or observations from the field, current model evaluation practices can be significantly improved through careful scrutiny of data and model functioning with CI methods. Two examples are the MC approach presented in Chapters 4 and 5, and the diagnostic evaluation approach of Chapter 6. Both these approaches are meant to extract information from data and model that is commonly ignored when merely evaluating the difference between time series of model output and observations using a single statistic. New methods of comparing signatures of model and data such as these are valuable because they signify an improved ability to judge the quality of hypotheses about the real-world hydrological system on which the model is based. This ability is considered improved not only because it is more accurate and reliable but ultimately a diagnostic tool through which one can find how hypotheses can be improved.

7.2 Recommendations

- Considering (1) the rapid advances in the field of CI, and (2) the diversity of techniques presented in this field, a most obvious recommendation is to keep exploring the effectiveness of CI techniques for application to R–R modeling. Any of the techniques mentioned in Chapter 2 can be used for application to hydrological modeling, and there are (and there will be) many more suitable ones.
  - As discussed in Chapter 2, many other variations on the traditional ANN exist (e.g., recurrent networks, radial basis function networks, neuro-fuzzy neural networks, wavelet neural networks, the spiking neural networks [Bothé, 2003]). Also, several alternatives to ANN data-driven models exist such as Support

Vector Machines and Genetic Programming models, which could shed more light on the value of the data-driven approach to R–R modeling.

  - Much progress has been made in the field of optimization over recent years. With the increase in computational power and the development of, for example, algorithms based on swarm intelligence (see Chapter 2), memetic algorithms [Hart *et al.*, 2005] and multialgorithms [Vrugt and Robinson, 2007], the possibilities for complex model optimization have grown. High-dimensional calibration problems such as ANN training or distributed R–R model calibration can benefit from the application of such sophisticated algorithms. A way to even more thoroughly explore the application of CI parameter estimation methods to R–R modeling would be to simultaneously estimate optimal parameter values and their probability distributions.
  - The k-means algorithm that was used for clustering is a relatively simple one. Examples of more powerful and informative alternatives are fuzzy clustering algorithms, self-organizing maps and the more recently developed random forests [Breiman, 2001].

- Proper development, calibration and evaluation of ANN R–R models continues to be a complex and opaque field. Much more insight in the effects of, for example, ANN model structures, objective functions, optimization algorithms, and initialization methods is needed before the use of data-driven techniques such as ANNs can either be recommended or discouraged as adequate alternatives to traditional R–R models.
- Hybrid methods that combine the best of data-driven and knowledge-driven models are valuable because they theoretically complement each other well. More research is suggested after how to optimally accomplish such combinations.
- The step of translating the diagnostic information found by data mining in Chapter 6 into structural model improvements is far from trivial. An approach that can find such translations in an objective way would be very valuable to hydrological modeling. More research on this topic is therefore suggested.

Appendix A

Study Sites and Data

A.1 Geer River Basin

The Geer River basin (Figures A.1 and A.2) is located in the north of Belgium, North West Europe, and is a subbasin of the Meuse River basin. The basin area is 494 km^2, and its mean annual rainfall is approximately 810 mm. The perennial river has discharges ranging from 1.8 m^3/s in dry periods to peaks of around 15 m^3/s.

Daily time series of rainfall at stations Waremme, Bierset and Visé, potential evaporation at Bierset, groundwater levels at Viemmes, and streamflow at the catchment outlet at Kanne were available for the periods 1980–1991 and 1993–1997. Areal rainfall was calculated from the three rainfall time series using the Thiessen polygon method. The time series for the two periods are connected into one time series, and the continuity of the data is largely preserved because the second period starts under similar flow conditions, and at the same point in the hydrological year where the first period ends. Hourly time series of rainfall at station Bierset and streamflow at Kanne were available for the period 1993–1997.

These data were made available by the Royal Meteorological Institute of Belgium (Brussels, Belgium).

A.2 Leaf River Basin

The Leaf River basin is located north of Collins, Mississippi, USA and has an area of approximately 1944 km^2. Time series of mean daily streamflow, daily potential evaporation estimates, and 6-hourly mean areal precipitation totals were available for the period 1948 to 1988. Mean annual precipitation of the basin is around 1400 mm and annual runoff around 400 mm.

These data were made available by the National Weather Service Hydrology Laboratory (MD, USA).

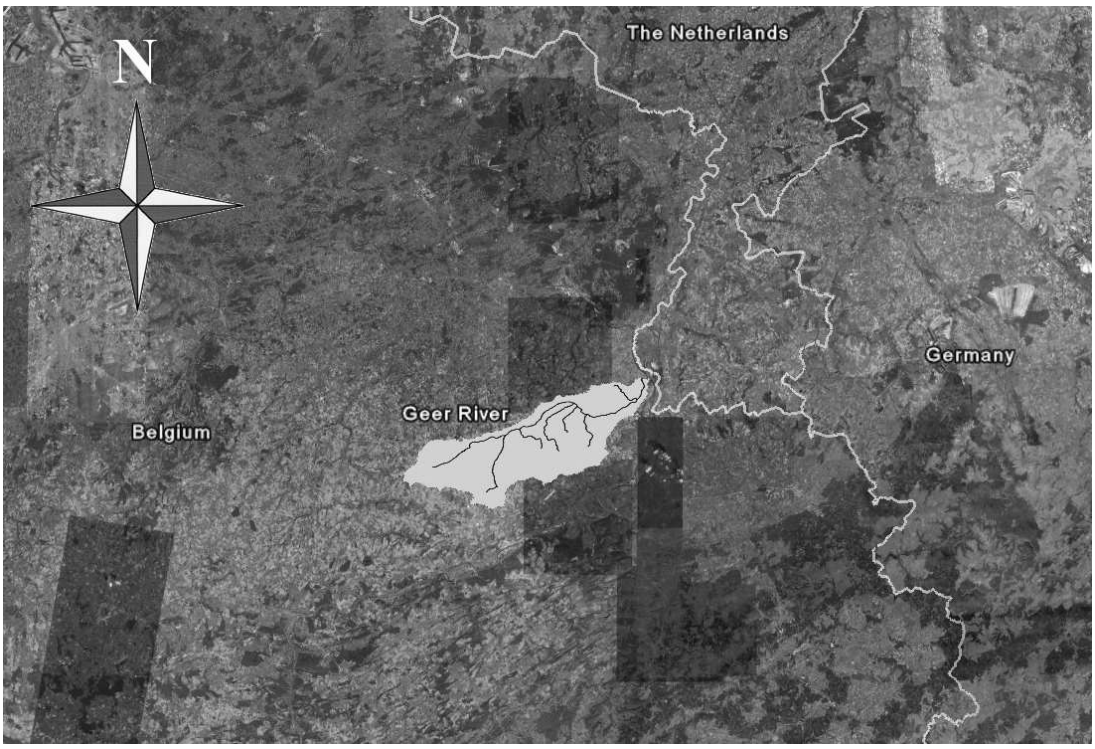


Figure A.1: Location of the Geer basin. *(Source: Google Earth.)*

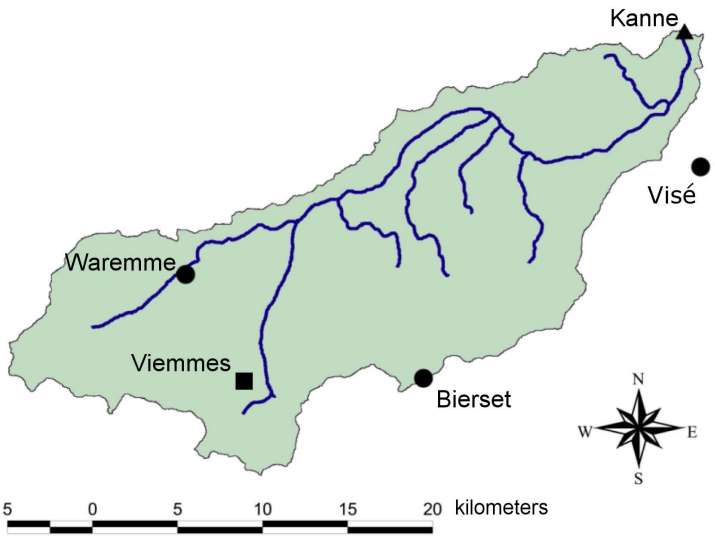


Figure A.2: Map of the Geer River basin, showing various measurement stations.

Figure A.3: Location of the Leaf River basin. *(Source: Google Earth.)*

References

Abbass, H. A. (2003). Speeding up backpropagation using multiobjective evolutionary algorithms. *Neural Comput.*, *15*, 2705–2726.

Abbott, M. B., Bathurst, J. C., Cunge, J. A., O'Connell, P. E., and Rasmussen, J. (1986a). An introduction to the European Hydrological System–Système Hydrologique Européen, SHE. 1. History and philosophy of a physically-based, distributed modelling system. *J. Hydrol.*, *87*, 45–59.

Abbott, M. B., Bathurst, J. C., Cunge, J. A., O'Connell, P. E., and Rasmussen, J. (1986b). An introduction to the European Hydrological System–Système Hydrologique Européen, SHE. 2. Structure of a physically-based, distributed modelling system. *J. Hydrol.*, *87*, 61–77.

Abrahart, R. J., Heppenstall, A. J., and See, L. M. (2006). Neural network forecasting: timing errors and autocorrelation issues. In *Proc. seventh international conference on hydroinformatics* (pp. 709–716). Nice, France: Research Publishing.

Abrahart, R. J., Heppenstall, R. J., and See, L. M. (2007). Timing error correction procedure applied to neural network rainfall-runoff modelling. *Hydrolog. Sci. J.*, *52*(3), 414–431.

Abrahart, R. J., and See, L. M. (2000). Comparing neural network and autoregressive moving average techniques for the provision of continuous river flow forecasts in two contrasting catchments. *Hydrol. Process.*, *14*(11), 2157–2172.

Ahmad, S., and Simonovic, S. P. (2005). An artificial neural network model for generating hydrograph from hydro-meteorological parameters. *J. Hydrol.*, *315*, 236–251.

Albuquerque Teixeira, R. de, Braga, A. P., Takahashi, R. H. C., and Saldanha, R. R. (2000). Improving generalization of MLPs with multi-objective optimization. *Neurocomputing*, *35*, 189–194.

Anctil, F., Michel, C., Perrin, C., and Andréassian, V. (2004). A soil moisture index as an auxiliary ANN input for stream flow forecasting. *J. Hydrol.*, *286*, 155–167.

Arthur, D., and Vassilvitskii, S. (2007). K-means++: the advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms* (pp. 1027–1035). New Orleans, LA, USA.

ASCE Task Committee on Application of Artificial Neural Networks in Hydrology. (2000). Artificial Neural Networks in hydrology, II: Hydrologic applications. *J. Hydrol. Eng.*, *5*(2), 124–137.

Asefa, T., Kemblowski, M., MacKee, M., and Khalil, A. (2002). Multi-time scale stream flow predictions: The support vector machines approach. *J. Hydrol.*, *318*, 7–16.

Babovic, V. (2005). Data mining in hydrology. *Hydrol. Process.*, *19*, 1511–1515.

Babovic, V., and Keijzer, M. (2002). Rainfall runoff modelling based on Genetic Programming. *Nordic Hydrol.*, *33*(5), 331–346.

Bárdossy, A., and Duckstein, L. (1995). *Fuzzy rule-based modeling with applications to geophysical, biological and engineering systems*. Boca Raton, FL, USA: CRC Press.

Bergström, S. (1976). *Development and application of a conceptual runoff model for Scandinavian catchments* (Tech. Report No. RHO 7). Norrköping, Sweden: Swedish Meteorological and Hydrological Institute (SMHI).

Betson, R. P., and Marius, J. B. (1969). Source areas of storm runoff. *Water Resour. Res.*, *5*(3), 574–582.

Beven, K. J. (1989). Changing ideas in hydrology — the case of physically based models. *J. Hydrol.*, *10*, 157–172.

Beven, K. J. (2001a). How far can we go in distributed hydrological modelling? *Hydrol. Earth Syst. Sc.*, *5*(1), 1–12.

Beven, K. J. (2001b). *Rainfall-runoff modelling: the primer*. Chichester, UK: John Wiley & Sons.

Beven, K. J. (2006). A manifesto for the equifinality thesis. *J. Hydrol.*, *320*, 18–36.

Beven, K. J., and Binley, A. M. (1992). The future of distributed models: model calibration and uncertainty prediction. *Hydrol. Proc.*, *6*, 279–298.

Beven, K. J., Lamb, R., Quinn, P. F., Romanowicz, R., and Freer, J. (1995b). TOPMODEL. In V. P. Singh (Ed.), *Computer models of watershed hydrology* (pp. 627–668). Colorado, USA: Water Resources Publications.

Bhaskar, N. R., and O'Connor, C. A. (1989). Comparison of method of residuals and cluster analysis for flood regionalization. *J. Water Resour. Plng. and Mgmt.*, *115*(6), 793–808.

Bothé, S. M. (2003). *Spiking neural networks*. Ph. D. dissertation, Leiden University, Leiden, The Netherlands.

Box, G. E. P., and Jenkins, G. M. (1976). *Time series analysis: Forecasting and control*. Holden Day.

Boyle, D. P. (2000). *Multicriteria calibration of hydrological models*. Ph. D. dissertation, Dep. of Hydrol. and Water Resour., Univ. of Ariz., Tucson, AZ, USA.

Boyle, D. P., Gupta, H. V., and Sorooshian, S. (2000). Towards improved calibration of hydrologic models: Combining the strengths of manual and automatic methods. *Water Resour. Res.*, *36*, 3663–3674.

Bray, M., and Han, D. (2004). Identification of support vector machines for runoff modelling. *J. Hydroinform.*, *6*, 265–280.

Brazil, L. E. (1988). *Multilevel calibration strategy for complex hydrologic simulation models*. Ph. D. dissertation, Colo. State Univ., Fort Collins, USA.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Brest, J., Greiner, S., Bošković, B., Mernik, M., and Žumer, V. (2006). Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Trans. Evol. Comput.*, *10*(6), 646–657.

Burn, D. H. (1989). Cluster analysis as applied to regional flood frequency. *J. Water Resour. Plng. and Mgmt.*, *115*(5), 567–582.

Burnash, R. J. C. (1995). The NWS river forecast system — catchment modeling. In V. P. Singh (Ed.), *Computer models of watershed hydrology* (pp. 311–366). Colorado, USA: Water Resources Publications.

Campolo, M., Andreussi, P., and Soldati, A. (1999). River flood forecasting with a neural network model. *Water Resour. Res.*, *35*(4), 1191–1197.

Chau, K. W. (2006). Particle swarm optimization training algorithm for ANNs in stage prediction of Shing Mun River. *J. Hydrol.*, *329*, 363–367.

Chen, S.-T., and Yu, P.-S. (2007). Pruning of support vector networks on flood forecasting. *J. Hydrol.*, *347*, 67–78.

Cheng, C. T., Ou, C. P., and Chau, K. W. (2002). Combining a fuzzy optimal model with a genetic algorithm to solve multi-objective rainfall–runoff model calibration. *J. Hydrol.*, *268*, 72–86.

Choi, H. T., and Beven, K. J. (2007). Multi-period and multi-criteria model conditioning to reduce prediction uncertainty in an application of TOPMODEL within the GLUE framework. *J. Hydrol.*, *332*, 316–336.

Conway, A. J., Macpherson, K. P., and Brown, J. C. (1998). Delayed time series predictions with neural networks. *Neurocomputing*, *18*, 81–89.

Cunge, J. A. (2003). Of data and models. *J. Hydroinformatics*, *5 (2)*, 75–98.

Dawson, C. W., See, L. M., Abrahart, R. J., and Heppenstall, A. J. (2006). Symbiotic adaptive neuro-evolution applied to rainfall–runoff modelling in northern England. *Neural Networks*, *19*(2), 236–247.

Dawson, C. W., and Wilby, R. L. (1999). A comparison of artificial neural networks used for river flow forecasting. *Hydrol. Earth Syst. Sc.*, *3*, 529–540.

Dawson, C. W., and Wilby, R. L. (2001). Hydrological modelling using artificial neural networks. *Prog. Phys. Geog.*, *25*, 80–108.

de Vos, N. J. (2003). *Rainfall–runoff modelling using artificial neural networks*. M. Sc. thesis, Delft University of Technology, Delft, The Netherlands.

de Vos, N. J., and Rientjes, T. H. M. (2005). Constraints of artificial neural networks for rainfall–runoff modelling: Trade-offs in hydrological state representation and model evaluation. *Hydrol. Earth Syst. Sc.*, *9*, 111–126.

de Vos, N. J., and Rientjes, T. H. M. (2007). Multi-objective performance comparison of an artificial neural network and a conceptual rainfall–runoff model. *Hydrolog. Sci. J.*, *52*(3), 397–413.

de Vos, N. J., and Rientjes, T. H. M. (2008a). Correction of timing errors of artificial neural network rainfall–runoff models. In R. J. Abrahart, L. M. See, and D. P. Solomatine (Eds.), *Practical hydroinformatics*. Springer: Water Science and Technology Library.

de Vos, N. J., and Rientjes, T. H. M. (2008b). Multi-objective training of artificial neural networks for rainfall–runoff modeling. *Water Resour. Res.*, *44*, W08434.

de Vos, N. J., Rientjes, T. H. M., and Pfister, L. (2004). Groundwater levels as state indicator in rainfall–runoff modelling using artificial neural networks. In B. Makaske and A. van

Os (Eds.), *Proceedings NCR-days 2004* (pp. 70–73). Delft, The Netherlands: Netherlands Centre for River Studies.

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. Chichester, UK: John Wiley & Sons.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA–II. *IEEE Trans. Evol. Computation*, *6*(2), 182–197.

Dibike, Y. B. (2002). *Model induction from data – Towards the next generation of computational engines in hydraulics and hydrology*. Ph. D. dissertation, UNESCO–IHE, Delft, The Netherlands.

Dibike, Y. B., and Solomatine, D. P. (2001). River flow forecasting using artificial neural networks. *Phys. Chem. Earth (B)*, *26*(1), 1–7.

Dooge, J. C. I., and O'Kane, J. P. (2003). *Deterministic methods in systems hydrology*. Taylor and Francis.

Dorigo, M., and Stützle, T. (2004). *Ant colony optimization*. MIT Press.

Duan, Q., Gupta, V. K., and Sorooshian, S. (1992). Effective and efficient global optimization for conceptual rainfall–runoff models. *Water Resour. Res.*, *28*, 1015–1031.

Duan, Q., Sorooshian, S., Gupta, H. V., Rousseau, A., and Turcotte, R. (2003). *Calibration of watershed models* (Vol. 6). Washington, DC: American Geophysical Union.

Dunne, T., and Black, R. D. (1970a). An experimental investigation of runoff production in permeable soils. *Water Resour. Res.*, *6*, 478–490.

Dunne, T., and Black, R. D. (1970b). Partial area contributions to storm runoff in a small New England watershed. *Water Resour. Res.*, *6*, 1296–1311.

Edijatno, N., Nascimento, O., Yang, X., Makhlouf, Z., and Michel, C. (1999). GR3J: a daily watershed model with three free parameters. *Hydrolog. Sci. J.*, *44*(2), 263–277.

Everitt, B., Landau, S., and Leese, M. (2001). *Cluster analysis*. New York, USA: Oxford University Press.

Fenicia, F., Savenije, H. H. G., Matgen, P., and Pfister, L. (2006). Is the groundwater reservoir linear? Learning from data in hydrological modelling. *Hydrology and Earth System Sciences*, *10*, 139–150.

Fenicia, F., Savenije, H. H. G., Matgen, P., and Pfister, L. (2007a). A comparison of alternative multi-objective calibration strategies for hydrological modelling. *Water Resour. Res.*, *43*(3), W03434.

Fenicia, F., Savenije, H. H. G., Matgen, P., and Pfister, L. (2008). Understanding catchment behavior through stepwise model concept improvement. *Water Resour. Res.*, *44*, W01402.

Fenicia, F., Solomatine, D. P., Savenije, H. H. G., and Matgen, P. (2007b). Soft combination of local models in a multi-objective framework. *Hydrol. Earth Syst. Sc.*, *11*, 1797–1809.

Fletcher, R., and Reeves, C. M. (1964). Function minimization by conjugate gradients. *Comput. J.*, *7*, 149–154.

Franchini, M., and Galeati, G. (1997). Comparing several genetic algorithm schemes for the calibration of conceptual rainfall-runoff models. *Hydrolog. Sci. J.*, *42*(3), 357–379.

Frapporti, G., Vriend, P., and Gaans, P. F. M. van. (1993). Hydrogeochemistry of the

shallow dutch groundwater: Interpretation of the national groundwater quality monitoring network. *Water Resour. Res.*, *29*(9), 2993–3004.

Freeze, R. A. (1972a). Role of subsurface flow in generating surface runoff, 1. Base flow contributions to channel flow. *Water Resour. Res.*, *8*(3), 609–623.

Freeze, R. A. (1972b). Role of subsurface flow in generating surface runoff, 2. Upstream source areas. *Water Resour. Res.*, *8*(5), 1272–1283.

French, M. N., Krajewski, W. F., and Cuykendall, R. R. (1992). Rainfall forecasting in space and time using a neural network. *J. Hydrol.*, *137*, 1–31.

Gallagher, R. G. (1968). *Information theory and reliable communication.* New York, USA: Wiley.

Gaume, E., and Gosset, R. (2003). Over-parameterisation, a major obstacle to the use of artificial neural networks in hydrology? *Hydrol. Earth Syst. Sc.*, *7*(5), 693–706.

Gautam, M. R., Watanabe, K., and Saegusa, H. (2000). Runoff analysis in humid forest catchment with artificial neural network. *J. Hydrol.*, *235*, 117–136.

Giustolisi, O., and Simeone, V. (2006). Optimal design of artificial neural networks by a multi-objective strategy: groundwater level predictions. *Hydrolog. Sci. J.*, *51*(3), 502–523.

Goldberg, D. E. (2000). *Genetic algorithms in search, optimization, and machine learning.* Reading, MA, USA: Addison–Wesley–Longman.

Goswami, M., and O'Connor, K. M. (2007). Comparative assessment of six automatic optimization techniques for calibration of a conceptual rainfallrunoff model. *Hydrolog. Sci. J.*, *52*(3), 432–449.

Gupta, H. V., Sorooshian, S., and Yapo, P. O. (1998). Toward improved calibration of hydrologic models: Multiple and noncommensurable measures of information. *Water Resour. Res.*, *34*(4), 751–763.

Gupta, H. V., Wagener, T., and Liu, Y. (2008). Reconciling theory and observations: elements of a diagnostic approach to model evaluation. *Hydrol. Process.*, *22*, 3802–3813.

Hagan, M. T., and Menhaj, M. B. (1994). Training feedforward networks with the marquardt algorithm. *IEEE Trans. Neural Networks*, *5*(6), 989–993.

Halff, A. H., Halff, H. M., and Azmoodeh, M. (1993). Predicting from rainfall using neural networks. In *ASCE proceedings of engineering hydrology* (pp. 760–765).

Hall, M. J., and Minns, A. W. (1999). The classification of hydrologically homogeneous regions. *Hydrolog. Sci. J.*, *44*(5), 693–704.

Han, D., Kwong, T., and Li, S. (2007). Uncertainties in real-time flood forecasting with neural networks. *Hydrol. Process.*, *21*, 223–228.

Hart, W. E., Krasnogor, N., and Smith, J. E. (2005). *Recent advances in memetic algorithms.* Berlin: Springer.

Haykin, S. (1999). *Neural networks, a comprehensive foundation.* Upper Saddle River, NJ, USA: Prentice Hall.

Hecht-Nielsen, R. (1990). *Neurocomputing.* Reading, MA, USA: Addison-Wesley.

Herbst, M., and Casper, M. C. (2007). Towards model evaluation and identification using Self-Organizing Maps. *Hydrol. Earth Syst. Sci. Discuss.*, *4*, 3953–3978.

Hewlett, J. D., and Hibbert, A. R. (1963). Moisture and energy conditions within a sloping soil mass during drainage. *Journal of Geophys. Res.*, *68*(4), 1081–1087.

Hewlett, J. D., and Hibbert, A. R. (1967). Factors affecting the response of small watersheds to precipitation in humid areas. In W. E. Sooper and H. W. Lull (Eds.), *Forest hydrology* (pp. 275–290). Oxford: Pergamon press.

Hjemfelt, A. T., and Wang, M. (1993). Artificial neural networks as unit hydrograph applications. In *ASCE proceedings of engineering hydrology* (pp. 754–759).

Hogue, T. S., Sorooshian, S., Gupta, H. V., Holz, A., and Braatz, D. (2000). A multi-step automatic calibration scheme for river forecasting models. *J. Hydrometeor.*, *1*, 524–542.

Holland, J. H. (1975). *Adaptation in natural and artificial systems.* Ann Arbor, MI, USA: University Michigan Press.

Houck, C., Joines, J., and Kay, M. (1995). *A genetic algorithm for function optimization: a MATLAB implementation* (Tech. Rep. No. NCSU-IE TR 95-09).

Hsu, K.-L., Gupta, H. V., Gao, X., Sorooshian, S., and Imam, B. (2002). Self-organizing linear output map (SOLO): An artificial neural network suitable for hydrologic modeling and analysis. *Water Resour. Res.*, *38*(12), 1302.

Hsu, K.-L., Gupta, H. V., and Sorooshian, S. (1995). Artificial neural network modeling of the rainfall–runoff process. *Water Resour. Res.*, *31*(10), 2517–2530.

Huang, W., Xu, B., and Chan-Hilton, A. (2004). Forecasting flows in Apalachicola River using neural networks. *Hydrol. Process.*, *18*(13), 2545–2564.

Jain, A., and Indurthy, S. K. V. P. (2003). Comparative analysis of event-based rainfall-runoff modeling techniques — deterministic, statistical, and artificial neural networks. *J. Hydrol. Eng.*, *8*(2), 93–98.

Jain, A., and Srinivasulu, S. (2004). Development of effective and efficient rainfall–runoff models using integration of deterministic, real-coded genetic algorithms and artificial neural network techniques. *Water Resour. Res.*, *40*(4), W04302.

Jain, A., and Srinivasulu, S. (2006). Integrated approach to model decomposed flow hydrograph using artificial neural network and conceptual techniques. *J. Hydrol.*, *317*, 291–306.

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: a review. *ACM Computing Surveys (CSUR)*, *31*(3), 264–323.

Jakeman, A. J., and Hornberger, G. M. (1993). How much complexity is warranted in a rainfall-runoff model? *Water Resour. Res.*, *29*(8), 2637-2649.

Jin, Y., Okabe, T., and Sendhoff, B. (2004). Neural network regularization and ensembling using multi-objective evolutionary algorithms. In *IEEE congress on evolutionary computation* (pp. 1–8).

Jong, K. A. D. (2006). *Evolutionary computation: A unified approach.* Cambridge, MA, USA: MIT Press.

Kamp, R. G., and Savenije, H. H. G. (2007). Hydrological model coupling with ANNs. *Hydrol. Earth Syst. Sc.*, *11*, 1869–1881.

Karunanithi, N., Grenney, W. J., Whitley, D., and Bovee, K. (1994). Neural network for river flow prediction. *J. Comput. Civil Eng.*, *8*(2), 201–220.

Kashif Gill, M., Kaheil, Y. H., Khalil, A., McKee, M., and Bastidas, L. (2006). Multiobjective

particle swarm optimization for parameter estimation in hydrology. *Water Resour. Res.*, *42*, W07417.

Kennedy, J., and Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of the 1995 ieee international conference on neural networks* (Vol. 4, pp. 1942–1948). IEEE Press, 1995.

Khu, S. T., Liong, S.-Y., Babovic, V., Madsen, H., and Muttil, N. (2001). Genetic programming and its application in real-time runoff forecasting. *J. Am. Water Resour. As.*, *37*(2), 439–451.

Khu, S. T., and Madsen, H. (2005). Multiobjective calibration with Pareto preference ordering: an application to runoff model calibration. *Water Resour. Res.*, *41*(3), W03004.

Kirkby, M. J. (1978). *Hillslope hydrology*. New York, USA: Wiley-Interscience.

Kirkby, M. J. (1988). Hillslope runoff processes and models. *J. Hydrol.*, *100*, 315–339.

Kirkpatrick, S., Jr., C. D. G., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, *220*, 671–680.

Kitanidis, P. K., and Bras, R. L. (1980). Real-time forecasting with a conceptual hydrologic model, 2: applications and results. *Water Resour. Res.*, *16*(6), 1034–1044.

Klemeš, V. (1983). Conceptualization and scale in hydrology. *J. Hydrol.*, *65*, 1–23.

Klemeš, V. (1986). Operational testing of hydrological simulation models. *Hydrolog. Sci. J.*, *31*(1), 13–24.

Kuczera, G., Kavetski, D., Franks, S., and Thyer, M. (2006). Towards a bayesian total error analysis of conceptual rainfall-runoff models: Characterising model error using storm-dependent parameters. *J. Hydrol.*, *331*, 161–177.

Lauzon, N., Anctil, F., and Baxter, C. W. (2006). Clustering of heterogeneous precipitation fields for the assessment and possible improvement of lumped neural network models for streamflow forecasts. *Hydrol. Earth Syst. Sc.*, *10*, 485–494.

Lecce, S. A. (2000). Spatial variations in the timing of annual floods in the southeastern united states. *J. Hydrol.*, *235*, 151–169.

Lindström, G., Johansson, B., Persson, M., Gardelin, M., and Bergström, S. (1997). Development and test of the distributed HBV-96 hydrological model. *J. Hydrol.*, *201*, 272–288.

Liong, S. Y., Lim, W. H., Kojiri, T., and Hori, T. (2000). Advance flood forecasting for flood stricken Bangladesh with a fuzzy reasoning method. *Hydrol. Process.*, *14*(3), 431–448.

Liong, S. Y., and Sivapragasam, C. (2002). Flood stage forecasting with SVM. *J. Am. Water Resour. As.*, *38*(1), 173–186.

Liu, J., and Lampinen, J. (2002). On setting the control parameter of the differential evolution method. In *Proc. 8th int. conf. soft computing (MENDEL 2002)* (pp. 11–18).

Liu, Y. Q., and Gupta, H. V. (2007). Uncertainty in hydrological modeling: Towards an integrated data assimilation framework. *Water Resour. Res.*, *43*, W07401.

Lloyd, S. P. (1982). Least squares quantization in PCM. *IEEE Trans. Inform. Theory*, *28*(2), 129–136.

Madsen, H. (2000). Automatic calibration of a conceptual rainfallrunoff model using multiple objectives. *J. Hydrol.*, *235*, 276–288.

Marshall, L., Nott, D., and Sharma, A. (2007). Towards dynamic catchment modelling: a Bayesian hierarchical mixtures of experts framework. *Hydrol. Process.*, *21*, 847–861.

Mazvimavi, D. (2003). *Estimation of flow characteristics of ungauged catchments: Case study in Zimbabwe*. Ph. D. dissertation, Wageningen University, Wageningen, The Netherlands.

Minns, A. W. (1998). *Artificial neural networks as subsymbolic process descriptors*. Ph. D. dissertation, UNESCO–IHE, Delft, The Netherlands.

Minns, A. W., and Hall, M. J. (1996). Artificial neural networks as rainfall–runoff models. *Hydrolog. Sci. J.*, *41*(3), 399–417.

Mitchell, T. M. (1997). *Machine learning*. New York, USA: McGraw–Hill.

Møller, M. F. (1993). A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, *6*(4), 525–533.

Moore, R. J. (1985). The probability-distributed principle and runoff production at point and basin scales. *Hydrolog. Sci. J.*, *30*(2), 273–297.

Mulvaney, T. J. (1851). On the use of self-registering rain and flood gauges in making observations of the relations of rainfall and flood discharges in a given catchment. *Proceedings of the Institution of Civil Engineers of Ireland*, *4*, 19–31.

Nash, J. E., and Sutcliffe, J. V. (1970). River flow forecasting through conceptual models; Part I – a discussion of principles. *J. Hydrol.*, *10*, 282–290.

Nayak, P. C., Sudheer, K. P., and Ramasastri, K. S. (2005). Fuzzy computing based rainfall-runoff model for real time flood forecasting. *Hydrol. Process.*, *19*, 955–968.

Oudin, L., Andréassian, V., Mathevet, T., Perrin, C., and Michel, C. (2006). Dynamic averaging of rainfall-runoff model simulations from complementary model parameterizations. *Water Resour. Res.*, *42*, W07410.

Palit, A. K., and Popovic, D. (2005). *Computational intelligence in time series forecasting: Theory and engineering applications*. London, UK: Springer.

Perrin, C., Michel, C., and Andréassian, V. (2003). Improvement of a parsimonious model for streamflow simulation. *J. Hydrol.*, *279*, 275–289.

Rajurkar, M. P., Kothyari, U. C., and Chaube, U. C. (2004). Modeling of the daily rainfall–runoff relationship with artificial neural network. *J. Hydrol.*, *285*, 96–113.

Reggiani, P., and Rientjes, T. H. M. (2005). Flux parameterization in the representative elementary watershed approach: Application to a natural basin. *Water Resour. Res.*, *41*(4), W04013.

Reggiani, P., Sivapalan, M., and Hassanizadeh, S. M. (2000). Conservation equations governing hillslope responses: Exploring the physical basis of water balance. *Water Resour. Res.*, *36*(7), 1845–1863.

Rientjes, T. H. M. (2004). *Inverse modelling of the rainfall–runoff relation: a multi objective model calibration approach*. Ph. D. dissertation, Delft University of Technology, Delft, The Netherlands.

Rooij, A. J. F. van, Johnson, R. P., and Jain, L. C. (1996). *Neural network training using genetic algorithms*. World Scientific Publishing, River Edge, NJ, USA.

Rumelhart, D. E., and McLelland, J. L. (1986). *Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1.* MIT Press, Cambridge, MA, USA.

Sajikumar, N., and Thandaveswara, B. S. (1999). A non-linear rainfall-runoff model using an artificial neural network. *J. Hydrol.*, *216*, 32–55.

Samani, N., Gohari-Moghadam, M., and Safavi, A. A. (2007). A simple neural network model for the determination of aquifer parameters. *J. Hydrol.*, *340*, 1–11.

Savenije, H. H. G. (2009). "the art of hydrology". *Hydrol. Earth Syst. Sc.*, *13*, 157–161.

Seibert, J. (2000). Multi-criteria calibration of a conceptual runoff model using a genetic algorithm. *Hydrol. Earth Syst. Sc.*, *4*(2), 215–224.

Seibert, J., and McDonnell, J. J. (2002). On the dialog between experimentalist and modeler in catchment hydrology: use of soft data for multicriteria model calibration. *Water Resour. Res.*, *38*(11), 1241.

Sexton, R., Dorsey, R., and Johnson, J. (1998). Toward global optimization of neural networks: a comparison of the genetic algorithm and backpropagation. *Decision Support Syst.*, *22*, 171–185.

Shamseldin, A. Y. (1997). Application of a neural network technique to rainfall-runoff modelling. *J. Hydrol.*, *199*, 272–294.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical J.*, *27*(3), 379–423.

Shoemaker, C. A., Regis, R. G., and Fleming, R. C. (2007). Watershed calibration using multistart local optimization and evolutionary optimization with radial basis function approximation. *Hydrolog. Sci. J.*, *52*(3), 450–465.

Sivapalan, M., Blöschl, G., Zhang, L., and Vertessy, R. (2003b). Downward approach to hydrological prediction. *Hydrol. Process.*, *17*(11), 2101–2111.

Smith, J., and Eli, R. N. (1995). Neural-network models of rainfall–runoff process. *J. Water Resour. Plng. and Mgmt.*, *121*(6), 499–508.

Solomatine, D. P. (2005). Data-driven modeling and computational intelligence methods in hydrology. In M. G. Anderson (Ed.), *Encyclopedia of hydrological sciences* (pp. 293–306). John Wiley & Sons.

Solomatine, D. P., and Dulal, K. N. (2003). Model tree as an alternative to neural network in rainfall-runoff modelling. *Hydrolog. Sci. J.*, *48*(3), 399–411.

Sorooshian, S., and Gupta, V. K. (1983b). Automatic calibration of conceptual rainfall-runoff models: the question of parameter observability and uniqueness. *Water Resour. Res.*, *19*(1), 260–268.

Srinivas, V. V., Tripathi, S., Rao, A. R., and Govindaraju, R. S. (2008). Regional flood frequency analysis by combining self-organizing feature map and fuzzy clustering. *J. Hydrol.*, *348*, 148–166.

Srivastav, R. K., Sudheer, K. P., and Chaubey, I. (2007). A simplified approach to quantifying predictive and parametric uncertainty in artificial neural network hydrologic models. *Water Resour. Res.*, *43*, W10407.

Storn, R., and Price, K. (1997). Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optimiz.*, *11*, 341–359.

Sudheer, K. P., and Jain, A. (2004). Explaining the internal behaviour of artificial neural network river flow models. *Hydrol. Process.*, *18*, 833–844.

Tang, Y., Reed, P., and Wagener, T. (2006). How effective and efficient are multiobjective evolutionary algorithms at hydrological model calibration? *Hydrol. Earth Syst. Sc.*, *10*, 289–307.

Thiemann, M., Trosset, M., Gupta, H. V., and Sorooshian, S. (2001). Bayesian recursive parameter estimation for hydrological models. *Water Resour. Res.*, *37*, 2521–2535.

Thirumalaiah, K., and Deo, M. C. (2000). Hydrological forecasting using neural networks. *J. Hydrol. Eng.*, *5*(2), 180–189.

Tokar, A. S., and Johnson, P. A. (1999). Rainfall–runoff modeling using artificial neural networks. *J. Hydrol. Eng.*, *4*(3), 232–239.

Tokar, A. S., and Markus, M. (2000). Precipitation–runoff modeling using artificial neural networks and conceptual models. *J. Hydrol. Eng.*, *5*(2), 156–160.

Toth, E. (2009). Classification of hydro-meteorological conditions and multiple artificial neural networks for streamflow forecasting. *Hydrol. Earth Syst. Sc. Discuss.*, *6*, 897–919.

Toth, E., Brath, A., and Montanari, A. (2000). Comparison of short-term rainfall prediction models for real-time flood forecasting. *J. Hydrol.*, *239*, 132–147.

Vapnik, V. N. (1998). *Statistical learning theory*. John Wiley & Sons, NY, USA.

Varoonchotikul, P. (2003). *Flood forecasting using artificial neural networks*. Ph. D. dissertation, UNESCO-IHE, Lisse, The Netherlands.

Vernieuwe, H., Georgieva, O., Baets, B. D., Pauwels, V. R. N., Verhoest, N. E. C., and Troch, F. P. D. (2005). Comparison of data-driven Takagi–Sugeno models of rainfall–discharge dynamics. *J. Hydrol.*, *302*, 173–186.

Vesterstrøm, J., and Thomsen, R. (2004). A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems. In *Proc. IEEE congr. evolutionary computation* (pp. 1980–1987). Portland, OR, USA.

Vrugt, J. A., Diks, C. G. H., Gupta, H. V., Bouten, W., and Verstraten, J. M. (2005). Improved treatment of uncertainty in hydrologic modeling: Combining the strengths of global optimization and data assimilation. *Water Resour. Res.*, *41*, W01017.

Vrugt, J. A., Gupta, H. V., Bastidas, L. A., Bouten, W., and Sorooshian, S. (2003a). Effective and efficient algorithm for multiobjective optimization of hydrologic models. *Water Resour. Res.*, *39*(8), 1214.

Vrugt, J. A., Gupta, H. V., Bouten, W., and Sorooshian, S. (2003b). A Shuffled Complex Evolution Metropolis algorithm for optimization and uncertainty assessment of hydrologic model parameters. *Water Resour. Res.*, *39*(8), 1201.

Vrugt, J. A., and Robinson, B. A. (2007). Improved evolutionary optimization from genetically adaptive multimethod search. *Proc. Natl. Acad. Sci.*, *104*(3), 708–711.

Wagener, T., Boyle, D. P., Lees, M. J., Wheater, H. S., Gupta, H. V., and Sorooshian, S. (2001). A framework for the development and application of hydrological models. *Hydrol. Earth Syst. Sc.*, *5*, 13–26.

Wagener, T., McIntyre, N., Lees, M. J., Wheater, H. S., and Gupta, H. V. (2003b). Towards reduced uncertainty in conceptual rainfall-runoff modeling: dynamic identifiability analysis. *Hydrol. Process.*, *17*(2), 455–476.

Wang, Q. J. (1991). The genetic algorithm and its application to calibrating conceptual

rainfall-runoff models. *Water Resour. Res.*, *27*(9), 2467–2471.

Wang, W. (2006). *Stochasticity, nonlinearity and forecasting of streamflow processes*. Ph. D. dissertation, Delft University of Technology, Delft, The Netherlands.

Wang, W., Gelder, P. H. A. J. M. van, Vrijling, J. K., and Ma, J. (2005). Testing and modelling autoregressive conditional heteroskedasticity of streamflow processes. *Nonlinear Proc. Geoph.*, *12*, 55–66.

Wilby, R. L., Abrahart, R. J., and Dawson, C. W. (2003). Detection of conceptual model rainfall-runoff processes inside an artificial neural network. *Hydrolog. Sci. J.*, *48*(2), 163–181.

Xie, X. L., and Beni, G. (1991). A validity measure for fuzzy clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, *13*(8), 841–847.

Xiong, L., Shamseldin, A. Y., and O'Connor, K. M. (2001). A non-linear combination of the forecasts of rainfall–runoff models by the first-order Takagi-Sugeno fuzzy system. *J. Hydrol.*, *245*, 196–217.

Yapo, P. O., Gupta, H. V., and Sorooshian, S. (1998). Multi-objective global optimization for hydrologic models. *J. Hydrol.*, *204*, 83–97.

Young, P. C. (2001). Data-based mechanistic modelling and validation of rainfall-flow processes. In M. G. Anderson and P. D. Bates (Eds.), *Model validation: Perspectives in hydrological science* (p. 117-161). Chichester: John Wiley.

Young, P. C. (2003). Top-down and data-based mechanistic modelling of rainfall–flow dynamics at the catchment scale. *Hydrol. Process.*, *17*, 2195–2217.

Young, P. C., and Beven, K. J. (1994). Data-Based Mechanistic (DBM) modelling and the rainfall–flow nonlinearity. *Environmetrics*, *5*, 335–363.

Zadeh, L. A. (1994). Soft computing and fuzzy logic. *IEEE Software*, 48–58.

Zadeh, L. A. (1996). Fuzzy logic = computing with words. *IEEE Trans. on Fuzzy Systems*, *4*, 103–111.

Zealand, C. M., Burn, D. H., and Simonovic, S. P. (1999). Short term streamflow forecasting using artificial neural networks. *J. Hydrol.*, *214*, 32–48.

Zhang, G. P., and Savenije, H. H. G. (2005). Rainfall-runoff modelling in a catchment with a complex groundwater flow system: application of the representative elementary watershed (rew) approach. *Hydrol. Earth Syst. Sc.*, *9*, 243–261.

Zhang, G. P., and Savenije, H. H. G. (2006). Modelling subsurface storm flow with the Representative Elementary Watershed (REW) approach: application to the Alzette River Basin. *Hydrol. Earth Syst. Sc.*, *10*, 937–955.

Zijderveld, A. (2003). *Neural network design strategies and modelling in hydroinformatics*. Ph. D. dissertation, Delft University of Technology, Delft, The Netherlands.

Zitzler, E., and Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Trans. Evol. Comput.*, *3*, 257–271.

Summary

Computational intelligence in rainfall–runoff modeling

The transformation from precipitation over a river basin to river streamflow is the result of many interacting processes which manifest themselves at various scales of time and space. The resulting complexity of hydrological systems, and the difficulty to properly and quantitatively express the information that is available about them, determine the challenge of Rainfall–Runoff (R–R) modeling. Accurate and reliable R–R models, however, are important because they can be used for scientific hypothesis testing, or for making prediction that can improve the quality or effectiveness of decisions related to water management issues.

In recent years, Computational Intelligence (CI) has emerged as a promising field of research. It has increasingly found application in R–R modeling (see literature review in Chapter 2), and in the research community there exists a clear and urgent need to further investigate the application of CI techniques. The main objective of this research, therefore, was to use CI techniques in catchment-scale R–R modeling in order to find improved methods of developing and evaluating such models. Three fields of application are explored for these purposes: system identification, parameter estimation and data mining. In Chapter 3, a R–R model based on a well-known CI technique, an Artificial Neural Network (ANN) is developed. Some important issues regarding the development, calibration and performance of such models are highlighted and discussed. Chapter 4 deals with the application of evolutionary, multi-criteria algorithms to the calibration of ANN R–R models, along with a comparison with traditional single-criterion algorithms. A multi-criteria comparison of an ANN model and a conceptual hydrological model is subsequently presented in Chapter 5. In Chapter 6, a temporal clustering approach was employed to identify periods of hydrological similarity. The results were used to shown how the evaluation of a conceptual model can be improved to be more diagnostic in nature and how subsequent improvements to the model structure can be inferred.

The following summarizes the investigations on each of the three applications.

1. Artificial Neural Networks as Data-Driven Models
   In Chapters 3, 4 and 5, ANNs have been used as data-driven R–R models to explore if CI techniques can simulate the R–R transformation adequately and how

well they compare to conceptual hydrological models. The results show good model performance, but also show that the application of ANNs is not without problems, since they are quite sensitive to several subjective modeling choices. For example, the choice of model input, structure or training algorithm has a big influence in the accuracy and parameter uncertainty of the model (Chapters 3 and 4). Moreover, ANNs sometimes appear to be sensitive to timing errors (Chapter 3). With respect to conceptual models, ANNs show to be slightly better for short-term forecasting but their performance decreases with increased forecast lead times (see Chapter 5). All in all, the development, calibration and evaluation of ANN R–R models, and the underlying uncertainties involved, which are subject to ANN model structure, objective functions, optimization algorithms, initialization, *et cetera*, continues to be a complex and opaque field. More insight in these issues is needed before the use of data-driven techniques such as ANNs can either be recommended or discouraged as adequate alternatives to traditional R–R models.

2. Computationally Intelligent Parameter Estimation
   CI parameter estimation algorithms have been applied to calibration of both CI and conceptual models to test whether more information can be extracted from hydrological data and used to make better R–R models (see Chapters 4 and 5). Throughout this study, the sensitivity of results on the method of optimization are shown to be large. Whether it be in differences between various local algorithms (Chapter 3), between local and global algorithms, or between single-criterion and multi-criteria algorithms (both in Chapter 4), the choice of algorithm turns out to have large effects on model accuracy and uncertainty. MC algorithms such as the NSGA–II and MOSCEM–UA prove to be very valuable in R–R model calibration since they exploit the information in model and data in a better way than , making the models not only accurate but a lot more reliable (see Chapters 4 and 5). Their usefulness naturally depends on the choice of objective functions. In this work, a new objective function (the Mean Squared Derivative Error) was proposed that penalizes a model for errors regarding hydrograph timing and shape. It was shown to evaluate models in a uniquely different way compared to traditional objective functions. Finally, the powerful self-adaptive Differential Evolution algorithm was employed in Chapter 6 and showed to be effective in a model calibration procedure. Generally, it was concluded that CI parameter estimation methods are more effective compared to traditional techniques.

3. Hydrological Clustering
   In order to find and make use of dynamical patterns in hydrological data that are commonly ignored in model evaluation, data mining has been performed in Chapter 6. A temporal clustering approach based on the simple k-means clustering algorithm was successfully devised to partition the historical data into several peri-

ods of hydrological similarity. The parameter variability between the hydrologically similar periods was subsequently used to make diagnostic inferences leading to improvements in the proposed model structures. This diagnostic step represents a successful novel application of clustering techniques that addresses the challenging and fundamental hydrological issue of how to achieve improvements on the working model hypothesis.

The overall conclusion of this work is that applications of CI in R–R modeling show a lot of promise. CI techniques can be considered powerful thanks to, for example, their general effectiveness in dealing with nonlinearity (e.g., ANNs as R–R models) and high-dimensionality (e.g., effectiveness of CI parameter estimation). Moreover, CI techniques provide an alternative viewpoint on hydrological data and models that is strikingly different from traditional techniques, and as such is able to extract information hitherto overlooked. Nevertheless, some pitfalls of CI techniques presented themselves. Therefore, there are advantages to approaches in which models use the characteristics of both knowledge-driven and data-driven modeling. This work shows that by correctly combining both process knowledge, modeling experience and intuition, it is possible to forge new model development and evaluation methods that combine the best of both worlds.

This work has shown that even without the use of additional sources of information or observations from the field, current model evaluation practices can be significantly improved through careful scrutiny of data and model functioning with CI techniques. Two examples from this work include the MC approach presented in Chapters 4 and 5, and the diagnostic evaluation approach of Chapter 6. Both these approaches are meant to extract information from data and model that is commonly ignored when merely evaluating the difference between time series of model output and observations using a single statistic. New methods of comparing signatures of model and data such as these are valuable because they signify an improved ability to judge the quality of hypotheses about the real-world hydrological system on which the model is based. This ability is considered improved not only because it is more accurate and reliable but ultimately a diagnostic tool through which one can find how hypotheses can be improved.

Samenvatting

Rekenkundige intelligentie in neerslag–afvoermodellering

De transformatie van neerslag op een rivierstroomgebied naar afvoer in de rivier is het resultaat van de interactie van vele processen die plaatsvinden op verscheidene tijd- en ruimteschalen. De hieruit voortvloeiende complexiteit van hydrologische systemen, samen met tekortkomingen om de beschikbare informatie correct en kwantitatief uit te drukken, maken van neerslag–afvoermodellering een uiterst moeilijke taak. Nauwkeurige en betrouwbare neerslag–afvoermodellen zijn echter belangrijk omdat zij gebruikt kunnen worden voor het testen van wetenschappelijke hypothesen, en voor het verbeteren van de kwaliteit of effectiviteit van beslissingen omtrent watermanagementsproblemen.

Recentelijk is Rekenkundige Intelligentie (RI) uitgegroeid tot een veelbelovend onderzoeksveld. Het wordt meer en meer toegepast in neerslag–afvoermodellering (zie het literatuuroverzicht in hoofdstuk 2), en in de wetenschappelijke gemeenschap bestaat een duidelijke en urgente behoefte om de toepassingen van zulke RI-technieken verder te onderzoeken. Het belangrijkste doel van dit onderzoek was daarom het toepassen van RI-technieken in neerslag–afvoermodellering op stroomgebiedsschaal, zodat verbeterde methoden voor het ontwikkelen en evalueren van zulke modellen gevonden kunnen worden. Drie toepassingsgebieden zijn onderzocht om dit doel te bereiken: systeemidentificatie, parameterschatting en 'data mining'. In hoofdstuk 3 wordt een bekende RI-techniek genaamd Kunstmatig Neuraal Netwerk (KNN) gebruikt om een neerslag–afvoermodel te ontwikkelen. Enkele belangrijke zaken omtrent het bouwen, het kalibreren en de prestaties van zulke modellen worden uitgelicht en bediscussieerd. Hoofdstuk 4 presenteert de toepassing van evolutionaire multi-criteria-algoritmes voor het kalibreren van KNN neerslag–afvoermodellen, met daarbij een vergelijking met algoritmes gebaseerd op een enkel criterium. In hoofdstuk 5 wordt vervolgens een multi-criteria-vergelijking tussen een KNN en een conceptueel neerslag–afvoermodel behandeld. In hoofdstuk 6 wordt clustering in de tijdsdimensie gebruikt om periodes van gelijkwaardige hydrologische toestand te identificeren. Deze informatie wordt vervolgens gebruikt om te laten zien hoe de evaluatie van een conceptueel model meer diagnostisch van aard kan worden gemaakt en hoe dit kan leiden tot verbeteringen van de modelstructuur.

Hieronder wordt het onderzoek aan de drie toepassingsgebieden samengevat.

1. Kunstmatige Neurale Netwerken als neerslag–afvoermodellen
   In hoofdstukken 3, 4 en 5 worden KNN gebruikt als data-gestuurde neerslag–afvoermodellen om te onderzoeken of RI-technieken de neerslag–afvoertransformatie adequaat kunnen simuleren en hoe goed ze zijn vergeleken met conceptuele hydrologische modellen. The resultaten laten goede prestaties zien, maar ook dat de toepassing van KNN niet zonder problemen is: ze zijn behoorlijk gevoelig ten opzichte van enkele subjectieve modelkeuzes. De keuze van modelinput, -structuur en trainingsalgoritme heeft een groot effect op de nauwkeurigheid en de parameteronzekerheid van het model (hoofdstukken 3 en 4). Ook blijken KNN-modellen gevoelig te zijn voor fouten in timing (hoofdstuk 3). In vergelijking met conceptuele modellen laten KNN betere korte-termijnresultaten zien, maar naarmate de voorspellingshorizon toeneemt, nemen hun prestaties af (hoofdstuk 5). Over het geheel genomen is de ontwikkeling, kalibratie en evaluatie van KNN neerslag–afvoermodellen, alsmede de daarmee samenhangende onzekerheden (die afhangen van bijvoorbeeld KNN modelstructuur, doelfuncties, optimalisatiealgoritmes en initialisatie), een complex en ondoorzichtig veld. Er is meer inzicht nodig voordat KNN aanbevolen of afgeraden kunnen worden als adequate alternatieven voor traditionele neerslag–afvoermodellen.

2. Rekenkundig Intelligente parameterschatting
   RI parameterschattingsalgoritmes zijn toegepast op de kalibratie van zowel RI- als conceptuele modellen om te testen of er zo meer informatie uit hydrologische data gehaald kan worden (hoofdstukken 4 en 5). De resultaten in deze gehele studie laten zien dat de gevoeligheid van resultaten voor de keuze van optimalisatiemethode groot zijn. Of het nu de verschillen tussen verschillende lokale algoritmes zijn (hoofdstuk 3), tussen lokale en globale algoritmes, of tussen enkel-criterium- en multi-criteria-algoritmes (beide in hoofdstuk 4), the algoritmekeuze heeft een groot effect op modelnauwkeurigheid en -onzekerheid. Multi-criteria-algoritmes zoals NSGA–II en MOSCEM–UA laten hun waarde zien in de kalibratie van neerslag–afvoermodellen omdat zij de informatie in model en data beter benutten dan enkel-criterium-algoritmes. De modellen worden daardoor niet alleen nauwkeuriger, maar ook betrouwbaarder (hoofdstukken 4 en 5). Het nut van deze algoritmes hangt natuurlijk wel af van de doelfuncties die gebruikt worden. In dit werk wordt een nieuwe doelfunctie voorgesteld die een model straft voor fouten in de timing en de vorm van het afvoerverloop. De beoordeling door deze functie verschilt op unieke wijze van traditionele doelfuncties. Ten slotte is het krachtige 'self-adaptive' Differential Evolution-algoritme gebruikt in hoofdstuk 6, waarbij het liet zien effectief te zijn voor modelkalibratie. Over het algemeen wordt geconcludeerd dat RI parameterschatting krachtiger is dan traditionele technieken.

3. Hydrologische clustering

Teneinde dynamische patronen in hydrologische data te vinden, en gebruik te maken van deze vaak genegeerde informatie, is 'data mining' uitgevoerd in hoofdstuk 6. Er is een succesvolle clustering in de tijdsdimensie uitgevoerd met behulp van het eenvoudige 'k-means' clusteringalgoritme, zodat de hydrologische data in verschillende perioden van gelijke hydrologische toestand kon worden verdeeld. De variabiliteit in parameters tussen deze periodes is vervolgens gebruikt om volgens de diagnostische methode verbeteringen aan de modelstructuur af te leiden. Deze diagnostische stap betekent een succesvolle nieuwe toepassing van clusteringtechnieken, welke het moeilijke en fundamentele hydrologische probleem aanpakt van hoe een bestaande modelhypothese te verbeteren.

De algemene conclusie van dit werk is dat de toepassing van RI in neerslag–afvoermodellering erg veelbelovend is. RI-technieken zijn krachtig vanwege bijvoorbeeld het feit dat zij effectief omgaan met niet-lineariteit (bijv. KNN als neerslag–afvoermodellen) en hogere dimensies (bijv. de effectiviteit van RI parameterschatting). Daarnaast benaderen RI-technieken hydrologische data en modellen vanuit een totaal andere richting dan traditionele technieken en daardoor kunnen zij informatie ontdekken die tot nu toe gemist werd. Niettemin blijkt de toepassing van CI-technieken enkele problemen te herbergen. Er hangen daarom voordelen aan benaderingen die de eigenschappen van zowel kennis-gestuurde als data-gestuurde modellen verenigen. Dit proefschrift laat zien dat door het correct combineren van proceskennis, modelleerervaring en intuïtie, het mogelijk is om nieuwe methoden voor modelontwikkeling en -evaluatie te smeden die het beste van beide werelden combineren.

Dit werk heeft laten zien dat, zelfs zonder het gebruik van nieuwe informatiebronnen of veldobservaties, huidige methoden voor modelevaluatie significant verbeterd kunnen worden door het nauwkeurig bestuderen van data en model met RI-technieken. Twee voorbeelden hiervan zijn de multi-criteria-methode in hoofdstukken 4 en 5, en de diagnostische evaluatiemethode van hoofdstuk 6. Beide methoden beogen informatie uit de data en het model te halen die normaal gesproken over het hoofd wordt gezien wanneer het verschil tussen gesimuleerde en geobserveerde tijdseries in een enkele waarde uitgedrukt wordt. Zulke innovatieve methoden zijn belangrijk omdat zij op nieuwe mogelijkheden duiden in het beoordelen van de kwaliteit van een hypothese van een stroomgebiedssysteem, waarop het model is gebaseerd. Deze mogelijkheden zijn niet alleen nauwkeuriger en betrouwbaarder, maar zelfs een diagnostisch hulpmiddel waarmee mogelijke verbeteringen aan de hypothese kunnen worden gevonden.

Curriculum Vitae

Nelis Jacob (Nico) de Vos was born on October 14, 1977 in Gorinchem, The Netherlands. After completing secondary school in 1996, he pursued studies in Civil Engineering at the Faculty of Civil Engineering and Geosciences of Delft University of Technology. Here he received his B. Sc. degree in 2000 and his M. Sc. degree in 2003, with a major in Hydrology and Ecology and a minor in Civil Engineering Informatics. His M. Sc. thesis title was 'Artificial Neural Networks for Rainfall–Runoff Modelling'.

In October 2003, Nico commenced his Ph. D. work at the Water Resources Section, Department of Water Management of Delft University of Technology on the subject of the application of computational intelligence techniques to rainfall–runoff modeling. From October to December 2006 he was a visiting researcher at the Hydrology and Water Resources Department of the University of Arizona in Tucson, USA. As of April 2008, he works for Transtrend B.V. in Rotterdam.