Numerical optimal control of open channel networks

From convex approximation to hidden invexity

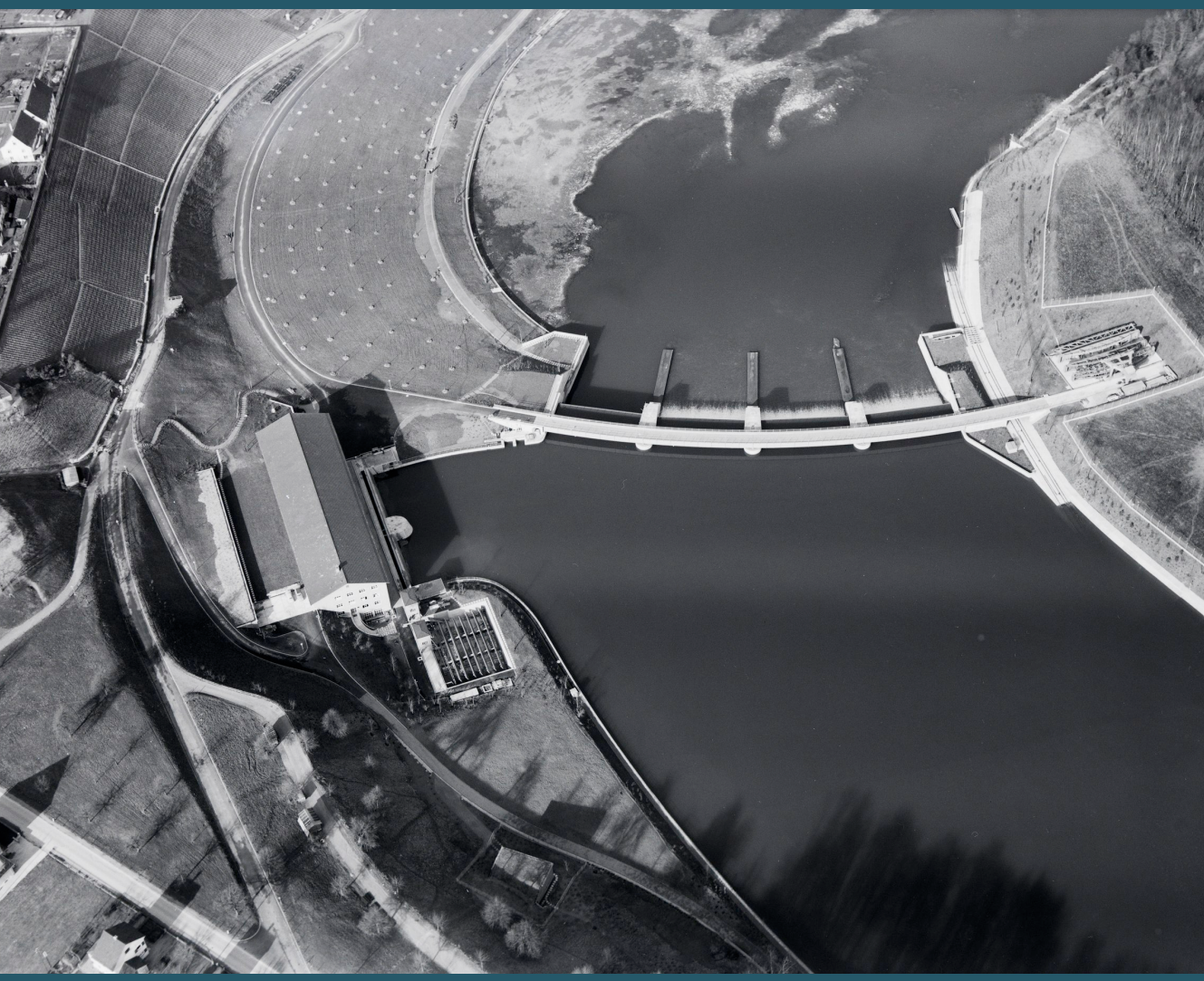
Baayen, J.H.

Publication date
2022

Document Version
Final published version

NUMERICAL OPTIMAL CONTROL OF OPEN CHANNEL NETWORKS

from convex approximation to hidden invexity



JORN H BAAYEN

Numerical optimal control of open channel networks

Academisch proefschrift

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. K.I.J. Maex
ten overstaan van een door het College voor Promoties
ingestelde commissie,
in het openbaar te verdedigen in de Aula der Universiteit
op woensdag 6 april 2022, te 14.00 uur

door Jorn Baayen
geboren te Amsterdam

Promotiecommissie

Promotor:

prof. dr. ir. D. den Hertog	Universiteit van Amsterdam

Copromotor:

dr. K. S. Postek	Technische Universiteit Delft

Overige leden:

prof. dr. M. Salomon	Universiteit van Amsterdam
prof. dr. S. I. Birbil	Universiteit van Amsterdam
prof. dr. J. A. S. Gromicho	Universiteit van Amsterdam
prof. dr. F. Liers	Friedrich-Alexander-Universität Erlangen-Nürnberg
prof. dr. ir. G. S. Stelling	Technische Universiteit Delft
prof. dr. D. A. Savić	University of Exeter

Faculteit Economie en Bedrijfskunde

Acknowledgements

First of all, I would like to thank my partner Mahtab, and my children Karel and Heide, for bearing with me through more than a few late-night research sessions.

I would like to thank Dr. Dirk Schwanenberg, author of the original RTC-Tools 1 and head of the Business Unit Water at KISTERS, for giving me the opportunity to continue my research at KISTERS. My gratitude also extends to my colleague Jesse VanderWees, who did a lot of the modelling work for some of the applications described in this dissertation.

I would like to thank all my collaborators at Deltares, my former colleagues, Dr. Bernhard Becker, Dr. Olav van Duin, Dr. Peter Gijsbers, Klaas-Jan van Heeringen, Dr. Klaudia Horváth, Judith ter Maat, Ivo Miltenburg, Dr. Teresa Piovesan, Simone van Schijndel, Jan Talsma, Dr. Matthijs den Toom, and Tjerk Vreeken. I would also like to thank Dr. Pierre Archambeau, prof. Vincenzo Casulli, Dr. Vyacheslav Kungurtsev, Dr. Jakub Mareček, prof. Jan van Schuppen, prof. Guus Stelling and prof. Ezio Todini for their insight and input.

My great-uncle, Harry Baayen, for his suggestion to pursue challenges in water management. My grandfather, the late prof. P. Cornelis Baayen, for inspiring me to pursue mathematics.

I would like to conclude by expressing my gratitude to prof. Dick den Hertog and Dr. Krzysztof Postek for their supervision and support.

Summary

Fresh water – if available in sufficient quantities – makes life possible. Too much water, on the other hand, can destroy property and end lives through flooding.

Throughout history, human civilizations have tried to stabilize nature's often erratic supply of fresh water by constructing dams to retain water in river valleys. In addition to retaining fresh water, such dams can reduce flooding by capturing flood wave peaks. Yet this can only work if sufficient retention *capacity* is available ahead of the flood wave's arrival. This leads to one of the central challenges of water management, both for the long term and in real-time: balancing the desire to retain water for future use with the need for retention capacity.

This balance is dynamic in time. In a dry season, it is possible to operate a reservoir at a high level as the risk of a flood event is low. In a wet season, on the other hand, a reservoir is typically operated at a lower level in order to ensure sufficient retention capacity. But if a normally wet season turns dry, such a static low level will become problematic: as water consumption continues and the reservoir is not replenished, water supply is at risk.

A natural evolution of the use of seasonal target water levels is real-time water management. In real-time water management, inflow forecasts are used to determine the optimal reservoir level dynamically. Decisions may be adjusted on a daily, hourly, or sometimes even quarter-hourly basis.

Real-time management of water resources hinges upon four factors: (1) an understanding of the water balance, *i.e.*, the inflows and extractions from the system, (2) the amount of storage available in a water system, (3) an understanding of wave propagation through river and canal, *i.e.*, open channel, systems, and (4) a

way to accomodate multiple, potentially conflicting, uses.

This dissertation concerns itself with the challenge of real-time multi-objective optimal control of open channel systems, *i.e.*, factors (3) and (4), while taking factors (1) and (2) into consideration.

The dynamics of water systems are nonlinear. Many real-time optimal control problems for water systems are therefore non-convex. From a mathematical point of view, this is problematic. In general, non-convex optimization problems admit an arbitrarily large number of local optima with different objective function values. Consequently, for such problems, the use of efficient local optimization methods is traditionally discouraged, and the use of expensive global optimization solvers, or of heuristic methods, is proposed instead.

In this dissertation, we address this non-convexity in two ways. The first approach is to accept a certain loss of model accuracy, and to approximate the non-convex problem with a convex problem. This is covered in Chapters 3 and 4.

The second approach is of a mathematically fundamental nature. This approach rests on a direct analysis of the quality of locally optimal solutions. The analysis is limited to a particular class of non-convex problems, for which it is proven that locally optimal solutions have strong nonlocal (global, or "nearly so") properties, a phenomenon referred to as "hidden invariant convexity". This analysis is covered in Chapters 5 and 6.

The hidden invariant convexity results of Chapter 5 apply to a class of discrete-time optimal control problems. This class covers, but is not limited to, problems arising in water resources management. The result is particularly important because it shows that for this class of non-convex problems, heuristics or compu-

tationally expensive global solvers are not needed, and that local, deterministic methods – such as first and second-order methods – are sufficient in practice.

The non-convex multi-objective approach has been succesfully applied for several real-world decision support systems. One of these systems, commissioned by the water authority of Rijnland in the *Randstad* area of the Netherlands, is described in Chapter 7.

Contents

1 Introduction **1**
1.1 Operational water management 1
1.2 Model predictive control 3
1.3 Discrete-time optimal control 4
1.4 Open-channel equations 5
1.5 Control structure equations 10
1.6 Optimization techniques 12
1.7 Contributions 14
1.8 Overview of chapters 17

I Convex approximation of flow network problems **21**

2 Quick Scan Tool for water allocation **23**
2.1 Introduction 23
2.2 Methods and techniques 28
2.3 The Quick Scan Tool application 34
2.4 Validation of model results 39
2.5 Discussion and conclusions 41

3 Convex modeling of pumps **43**
3.1 Introduction 43
3.2 Methodology 47

3.3 Results and discussion 62
3.4 Conclusions 70
3.A Calculation of pump speed using affinity rules 72

4 MPC of a river reach with weirs 75
4.1 Introduction 75
4.2 Material and methods 77
4.3 Case study 87
4.4 Conclusions 93
4.A Derivation of the boundaries of the working area 98

II Invex optimization of flow network problems 101

5 Hidden invexity in discrete-time optimal control 103
5.1 Introduction 103
5.2 Regular discrete-time optimal control problems 106
5.3 Hidden invexity 115
5.4 Numerical experiment: river water level control 125
5.5 Conclusions and outlook 133

6 Benchmark: Open channel water level control 135
6.1 Introduction 135
6.2 Benchmark setup 135
6.3 Results 137
6.4 Conclusions 145

7 Case study: Hoogheemraadschap van Rijnland 147
7.1 System description 147
7.2 Model setup 150
7.3 Results 151

CHAPTER 1

Introduction

We begin this introductory chapter by sketching current challenges and drivers in operational water management and explain how these lead to mathematical optimization (Section 1.1). In Section 1.2, we sketch the model predictive control (MPC) framework within which such optimization problems are operationalized. In Section 1.3, the overall structure of the optimization problems that arise in the context of an MPC controller is described. This structural description will have placeholders for mathematical models of flow in channel reaches (Section 1.4), and for the behaviour of the various types of control structures (Sections 1.5). Most of these models are nonlinear, which makes solving these optimization problems challenging in general. A variety of techniques rising to the challenge are described in Section 1.6. In Section 1.7, the core contributions of this thesis are presented. Section 1.8 provides an overview of the research papers contained in this thesis.

1.1 Operational water management

Open-channel flow is a type of liquid flow within a channel with a free surface; the other type of flow is pipe flow, where there is no

free surface. Examples of open-channel networks are natural river systems, as well as man-made canal systems such as those found in and around the low-lying, actively drained polder systems of the Netherlands. These open-channel networks are often endowed with flow control structures, such as weirs (overflow dams), gates, dams, and pumping stations, that enable operators to control water flows and levels. An example of an open-channel network is shown in Figure 1.1.

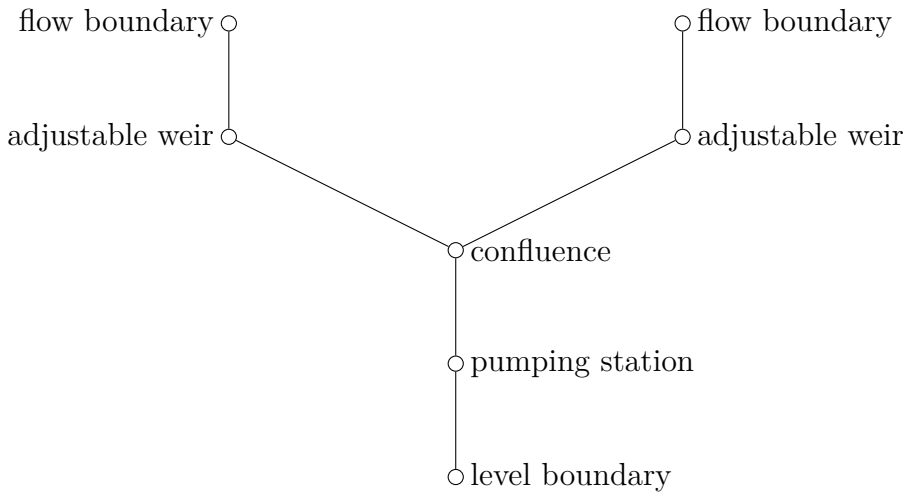


Figure 1.1 – Schematic overview of an open-channel network with two upstream inflow boundary conditions, two adjustable weirs, a confluence, a pumping station containing one or more pumps, and a downstream level boundary condition. The adjustable weirs and the pumping station are control structures. The lines represent the channel reaches.

Typically, flows and levels are regulated to satisfy several, potentially conflicting, requirements. These requirements may include, for example, (a) water availability for agriculture, industry, and

drinking water production, (b) flood control, (c) ecosystem health, (d) stable water levels for recreation, (e) operation of pumps at minimal cost, and (f) operations of hydroelectric turbines for maximal energy generation or energy sales revenue.

Optimally satisfying a combination of such requirements is a complex task. Traditionally, organizations rely on the expertise of human operators (van der Zwan 2017, e.g.). In recent years, various stresses have been driving water authorities to look for alternative ways to operate their systems. Examples of such stresses include (a) climate change driving an increased prevalence of extreme events, (b) tightened environmental regulation requiring water system operators to operate more carefully, (c) budget cuts causing water agencies to look for ways to reduce expenditures, and (d) bans on the construction of new dams causing hydropower companies to look for alternative ways to grow revenue. Such stresses, as well as a general desire to innovate, drive water management agencies towards the adoption of mathematical optimization techniques.

1.2 Model predictive control

Model predictive control (MPC) is a popular control technique that uses optimization to compute a dynamically evolving control strategy (García et al. 1989). At every controller time step, a discrete-time optimal control problem is solved over a finite time horizon. The *here-and-now decisions*, *i.e.*, the decisions of first time step of the computed control strategy, are implemented in practice. At the next controller time step, new measurements are recorded, the time period is shifted, and the process is repeated. MPC has been studied extensively for water systems (Ackermann et al. 2000, van Overloop 2006a, Litrico and Fromion 2009, Schwa-

nenberg et al. 2015, e.g.).

1.3 Discrete-time optimal control

Given an open-channel network of channel reaches $\mathcal{R}$ and control structures $\mathcal{S}$, the discrete-time optimal control problems arising in an MPC controller have the following general scheme:

$$\min_{x,u} \; f(x,u) \tag{1.1}$$

$$\text{s.t. } c^r_{t=t_j}\left(x(t_{j-1}), x(t_j), u(t_{j-1}), u(t_j)\right) = 0 \quad j = 1,\ldots,T,\ r \in \mathcal{R},$$

(open-channel equations)

$$d^s_{t=t_j}\left(x(t_{j-1}), x(t_j), u(t_{j-1}), u(t_j)\right) = 0 \quad j = 1,\ldots,T,\ s \in \mathcal{S},$$

(control structure equations)

$$\underline{u} \le u(t_j) \le \overline{u} \qquad j = 1,\ldots,T,$$

where $u(t_j)$ represents the vector of control decisions at time t_j, and $\underline{u}$ and $\overline{u}$ its lower and upper bounds, respectively. The vector x represents the optimization variables that are not control decisions, but that arise in the equations of the open-channel model. We will refer to these variables as *states*. The vector of states at time t_j is denoted by $x(t_j)$. The vector-valued function $c^r_{t=t_j}$ represents the open-channel equations for reach r at time step t_j, and the vector-valued function $d^s_{t=t_j}$ represents the control structure equations for structure s at time step t_j. The objective function f is problem-specific, and is generally chosen to be a jointly convex function of the states x and the controls u. Problem (1.1) contains placeholders for equations describing flow in the open-channel reaches and describing the behaviour of the control structures. In Sections 1.4 and 1.5, we will discuss a variety of ways to model open-channel reaches and control structures, respectively.

Optimization problems for water systems generally involve multiple objective functions f. In an MPC setting, it is generally

undesirable to trade these objectives off against each other, as the weighting factors for the various objectives would generally be hard to interpret. Experience shows that such opaque parameters require constant tuning and erode the confidence of the operators of a water system.

An established approach for multi-objective optimization in water systems is *lexicographic goal programming* (LGP) (Collette and Siarry 2003, Eschenbach et al. 2001). In LGP, optimization objectives are ordered according to priority. The problem is then solved for different objectives in lexicographic order. Following every optimization run, a constraint is added that fixes the attained objective value. In this way, the optimization of subsequent objectives cannot decrease the values of the objectives that have already been optimized for. This approach is described in detail in Chapter 2.

Realistic problems typically contain hundreds or thousands of control variables, thousands to hundreds of thousands of state variables, and an equal number of equality constraints. These numbers depend on the size of the network, the resolution of the spatial discretization (when using hydrological or hydraulic channel models), and the number of time steps.

1.4 Open-channel equations

In optimization problem (1.1), the channel reaches (cf. Figure 1.1) are represented using equality constraints. In the following, we cover the three most common models that can be used to provide these equalities, in increasing order of physical detail.

Integrator-delay models. The integrator-delay model is a simple way to model flow propagation in a controlled water system. It arises out of the observation that water in an open-channel

takes a finite amount of time to travel from one control structure to the next. Whereas in reality the relationship between the flow velocity and the water level is nonlinear, the integrator-delay model assumes that the water moves with a certain fixed velocity. It does not capture any other physical phenomena, such as wave diffusion.

The integrator-delay model is given by the equation (Falk et al. 2016, e.g.):

$$\frac{dV}{dt}(t) = Q_{in}(t) - Q_{out}(t + \tau),$$

with reach volume V, time t, upstream inflow Q_{in} and downstream outflow Q_{out}. This equation decribes the conservation of volume, and delays the downstream outflow by the time offset τ. The function of the variables is illustrated in Figure 1.2.

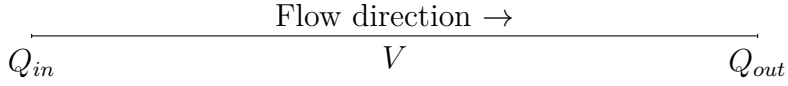


Figure 1.2 – Conceptual view of an open-channel reach.

Continuous-time variants of the integrator-delay model exist as well. The *integrator-delay-zero* model of Litrico and Fromion (2004), for example, is a continuous-time linear time-invariant model that introduces a (frequency-dependent) phase delay to the outflow Q_{out}.

The integrator-delay model is appropriate for water allocation applications such as the one described in Chapter 2, where short-term hydrodynamic effects such as wave propagation and diffusion are largely irrelevant. It is not appropriate for applications that require damping of flood waves. Nor is it appropriate for the timing of hydroelectric power generation along a cascade of run-of-river power plants, where water released during generation

peaks travels as a wave from one plant to the next.

Hydrological flow routing. In natural rivers, it is common to observe an empirical correlation between flow Q and the volume V stored in a reach. When modelling a reach as having an inflow Q_{in} and an outflow Q_{out}, as in Figure 1.2, the reach flow Q may be expressed as a weighted average of the in- and outflows, *i.e.*, $Q = XQ_{in}+(1-X)Q_{out}$. The empirical correlation is expressed as the so-called *Muskingum flow routing equation* (McCarthy 1938, e.g.):

$$V = K\left(XQ_{in} + (1-X)Q_{out}\right),$$

with parameters $K > 0$ and $0 \leq X \leq 1$. In the classical Muskingum approach, K and X are taken fixed and are fitted to available measurement data. The Muskingum flow routing equation is typically accompanied by a *mass balance equation* of the form

$$\frac{dV}{dt} = Q_{in} - Q_{out}.$$

Reach confluences are modelled by summing the downstream outflows of upstream reaches, and feeding the sum of outflows as an upstream inflow to the downstream reach.

The approach was later extended by Cunge, who derived equations for K and X from a hydraulic analysis (Cunge 1969, Todini 2007, e.g.):

$$K = \frac{\Delta x}{c},$$
$$X = \frac{1}{2}\left(1 - \frac{Q_{out}}{BS_0c\Delta x}\right),$$

with wave celerity c, reach length Δx, channel width B, and channel bed slope S_0. Whereas the classical Muskingum method results in a linear model, the generally more accurate Muskingum-Cunge model has bilinear terms $Q_{in}Q_{out}$ and Q_{out}^2.

Hydrological routing schemes are typically appropriate when (a) the flow is always in one direction, and (b) there are no *backwater effects* to speak of, *i.e.*, water levels do not build up due to a downstream flow restriction. As such, hydrological routing is often an appropriate choice when modelling natural rivers that flow downhill. It is, however, not suitable for flow that changes direction depending on relative water levels, as is the case for many flat canal systems in the Netherlands. It is also generally not suitable to model run-of-river hydropower schemes, where a plant may restrict the flow and cause water levels upstream of the plant to rise.

Hydraulic models. At a fundamental level, fluid flow is described by the Navier-Stokes equations. Using the assumptions that water is incompressible and that the pressure is hydrostatic, *i.e.*, proportional to the weight of the column of water above an infinitesimally small control volume, the Navier-Stokes equations can be simplified to obtain the *shallow water equations*. In their most basic form, the shallow water equations describe flow in open channels and shallow lakes (Chanson 2004, Vreugdenhil 2013, e.g.).

By averaging variables in the vertical direction and in the horizontal direction orthogonal to the channel, the one-dimensional shallow water equations may be obtained. These are also known as the *Saint-Venant* equations, and are given by the *momentum equation*

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}\frac{Q^2}{A} + gA\frac{\partial H}{\partial x} + g\frac{Q|Q|}{ARC^2} = 0, \tag{1.2}$$

with the longitudinal coordinate x increasing in the flow direction of the river, time t, flow (discharge) Q, water level H, cross section A, hydraulic radius $R := A/P$, wetted perimeter P, Chézy friction coefficient C, gravitational constant g, and by the *mass balance*

(or *continuity*) equation

$$\frac{\partial Q}{\partial x} + \frac{\partial A}{\partial t} = 0. \tag{1.3}$$

These hyperbolic partial differential equations (PDE) describe the time evolution of the level and flow of water at each point of an open-channel reach. The Saint-Venant equations capture most physically relevant hydrodynamic phenomena, including dynamic wave propagation velocity, wave diffusion, and backwater effects. These equations can therefore be used to model cyclic networks, and networks where the flow direction is not fixed a priori, such as the drainage networks of the polders in the Netherlands.

A common simplification is to consider the momentum equation (1.2) in steady state, by setting the total derivative of the velocity $u := Q/A$ to zero:

$$0 = \frac{du}{dt} = \frac{d}{dt}\frac{Q}{A} = \frac{1}{A}\left(\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}\frac{Q^2}{A}\right),$$

where the continuity equation (1.3) is used to derive the last equality. In this way one obtains the *diffusive wave* equation:

$$gA\frac{\partial H}{\partial x} + g\frac{Q|Q|}{ARC^2} = 0.$$

An intermediate form, the *inertial wave* equation (Montero et al. 2013), is also used:

$$\frac{\partial Q}{\partial t} + gA\frac{\partial H}{\partial x} + g\frac{Q|Q|}{ARC^2} = 0. \tag{1.4}$$

The inertial wave equation is identical to the Saint-Venant momentum equation, except for the convective acceleration term $\partial/\partial x\,(Q^2/A)$, which is omitted. For subcritical flow, this term is small in magnitude relative to the others. This omission effects a reduction in computational effort.

In order to compute with the Saint-Venant equations or any of its simplifications, an approximation, discrete in time and space, is required. In this dissertation we will use the semi-implicit approximation scheme of Casulli and Cheng (1992). This discretization is detailed and illustrated with a numerical example in Chapter 5.

In our optimization problem (1.1), the (discretized) open-channel equations are included as equality constraints. As such, the nonlinear Saint-Venant equations produce a non-convex optimization problem. Some other nonlinear PDE, such as the Hazen-Williams or Darcy-Weisbach equations for pressurized flow in pipes, can be discretized as convex functions of the states. These convex functions may be included in an optimization problem as *inequality* constraints (Sela Perelman and Amin 2015, e.g.), thereby producing a convex relaxation of the original problem. The Saint-Venant equations, however, are not convex functions and therefore result in a non-convex optimization problem, regardless of whether they are included as equality or as inequality constraints. From an optimization perspective, the Saint-Venant equations are therefore fundamentally difficult to work with.

1.5 Control structure equations

In optimization problem (1.1), the control structures (cf. Figure 1.2) are also represented using equality constraints. We will give three examples of control structures and the equations describing their behaviour.

Weir equation. A weir is a small dam, generally set up such that there is a reduced flow of water over its crest. Some weirs are *adjustable*, *i.e.*, their crest level is controllable. Such a structure presents a means to regulate the flow.

The weir equation is

$$Q = CL(\Delta H)^n, \tag{1.5}$$

with hydraulic head difference

$$\Delta H = H_{up} - H_c,$$

and flow coefficient C, crest width L, and power n (Chanson 2004, e.g.). The power n varies with the structure. For a horizontal weir, $n = \frac{3}{2}$. The crest level H_c is the control variable; the upstream water level H_{up} is a state variable.

The weir equation is nonlinear.

Pump equation. The power consumption of a pump is a multiple of the hydraulic power required to produce a given flow, *i.e.*,

$$\eta(Q, H_{up}, H_{down})\, P = g\rho Q \Delta H, \tag{1.6}$$

with hydraulic head difference

$$\Delta H = H_{down} - H_{up},$$

and efficiency function η with $\eta(Q, H_{up}, H_{down}) \in [0, 1]$, power consumption P, gravitational constant g, water density ρ (constant), pump flow Q, upstream water level H_{up} and downstream water level H_{down} (Chanson 2004, e.g.). Either the power consumption P, or the pump flow Q, may be taken as control variable; the upstream and downstream water levels H_{up} and H_{down} are state variables.

If the efficiency η is taken constant, the power consumption P is bilinear in the flow Q and the hydraulic head ΔH.

Turbine equation. The power generation of a turbine is a fraction of the available hydraulic power, *i.e.*,

$$P = g\rho\eta(Q, H_{up}, H_{down})\, Q \Delta H,$$

with hydraulic head difference

$$\Delta H = H_{up} - H_{down},$$

and power generation P, gravitational constant g, water density ρ (constant), efficiency function η with $\eta(Q, H_{up}, H_{down}) \in [0, 1]$, pump flow Q, upstream water level H_{up} and downstream water level H_{down} (Chanson 2004, e.g.). Either the power generation P, or the turbine flow Q, may be taken as control variable; the upstream and downstream water levels H_{up} and H_{down} are state variables.

If the efficiency η is taken constant, the power generation P is bilinear in the flow Q and the hydraulic head ΔH.

1.6 Optimization techniques

In this section, we discuss various techniques for solving optimization problems of the form (1.1).

All of the models discussed above, *except* the integrator-delay model for flow routing, are nonlinear. Therefore, a direct inclusion of any of these models as equality constraints between optimization variables, would, in general, result in a non-convex optimization problem (Boyd and Vandenberghe 2004, e.g.). Non-convex optimization is, *in the general case*, of limited practical use as a local minimum need not be a global minimum.

A common "work-around" is to work with simplified models so as to obtain a linear or a convex optimization problem (Eschenbach et al. 2001, e.g.), for which every local minimum is a global minimum (Boyd and Vandenberghe 2004, e.g). For example, the integrator-delay model can be used, and the power consumption P of a pump may be modelled using a constant *energy coefficient*

E as

$$P = EQ, \tag{1.7}$$

with flow Q, *i.e.*, completely dropping the dependency on the hydraulic head – compare Equation (1.6).

Another approach is to linearize the nonlinear equations around a nominal point, and then optimize this linearized model (Amann et al. 2016, e.g.). Here, one makes the assumption that the states occuring in the optimization do not deviate "too much" from the nominal point. This assumption is violated for, e.g., flood waves, where flow increases and then decreases sharply. Iterative refinement through sequential linear programming is an option, but this is conceptually no different from local nonlinear optimization.

Other authors apply a general-purpose global solver such as COUENNE (Belotti et al. 2009), or apply Lasserre hierarchies (Lasserre 2001, Ghaddar et al. 2017, e.g.), in order to find solutions that are provably globally optimal. While such techniques eventually find a global optimum, the process may take hours, days, or worse (Ghaddar et al. 2017, e.g.). Such run times are not suitable for closed-loop MPC, where, for a typical water system, the controller is invoked at an hourly frequency.

Yet other authors have tried to address the difficulty of non-convex optimization by using genetic algorithms (Van Zyl et al. 2004, Nicklow et al. 2010, Vermuyten et al. 2018, e.g.). The solutions discovered using a genetic algorithm are, however, not guaranteed to be locally, let alone globally, optimal. Practical experiments show that substantial random variation exists in solutions obtained using such algorithms (Vermuyten et al. 2018, e.g).

In Schwanenberg et al. (2015), local nonlinear, non-convex optimization is applied directly. Similar non-convex approaches are applied for drinking water distribution networks in Burgschweiger

et al. (2009), and for gas transport networks in Hante et al. (2017). In all of these works, local nonlinear optimization is applied, but the optimality properties of the solutions are not studied. In Ghaddar et al. (2017), drinking water distribution problems are studied, and the solutions obtained by the local solver IPOPT (Wächter and Biegler 2006) are compared with global solutions computed using COUENNE on the one hand, and using Lasserre hierarchies on the other hand. In said work, an intriguing result is obtained: the solutions obtained using the local method are practically indistinguistable from the global solutions.

1.7 Contributions

Prior to the research described in this dissertation, there were no adequate analytical tools to analyze non-convex hydrodynamic optimization problems. Initially, this circumstance drove the convex approximation work described in the first part of this thesis. Subsequently, it drove the creation of the new analytical framework to analyze nonlinear water system optimization problems directly. The framework is based on the theory of invex (Hanson 1981, Craven 1981a, Ben-Israel and Mond 1986) functions, and is covered in the second part of this thesis.

The relationship between linear optimization, convex approximation, and invex optimization is illustrated in Figure 1.3.

Convex approximation. A contribution of this thesis is a more accurate convex approximation approach for the modelling of control structures. In Chapters 3 and 4, it is shown how the classical linear approximation (1.7) may be replaced by a more – but not fully – accurate convex approximation.

Invex optimization. The second – and primary – contribution of this dissertation is a principled approach towards the analysis

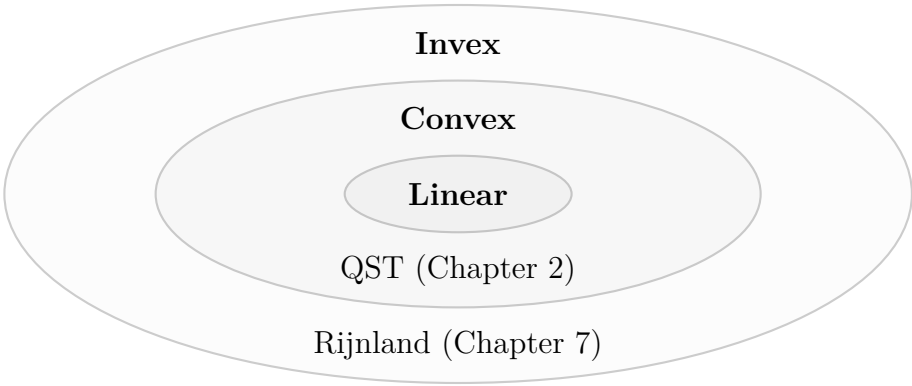


Figure 1.3 – Invex optimization comprises all of convex optimization, which contains all of linear optimization. The two operational applications described in this dissertation, the Quick Scan Tool (QST) and the MPC controller for Rijnland, are shown in their respective classes. The QST is either linear or convex depending on the p-norm chosen for its objective functions.

and solution of non-convex hydrodynamic optimization problems. In Chapter 5 it is shown how, under certain conditions satisfied by *appropriately discretized* Saint-Venant and *appropriately formulated* control structure equations, the objective becomes an invex function of the control variables in the original sense of Hanson (1981) and Craven (1981b). On an open set, invex functions have the property that every stationary point is a global minimum (Ben-Israel and Mond 1986, e.g.). At the boundary of the domain, tangent cones are used to prove global optimality over a "large" subset of the domain. We minted the term *conic-intersection optimality* to refer to this phenomenon.

For a broad class of non-convex hydrodynamic optimization problems, the result shows that local methods find solutions with conic-intersection optimality guarantees at worst, and full global opti-

mality at best. For these problems, *there is no need for Lasserre hierarchies, genetic algorithms, or other complex or heuristic techniques*, and local methods are sufficient for most practical purposes. It also obsoletes the need for the convex approximations of Chapters 3 and 4.

The new theory confirms the earlier computational results of Ghaddar et al. (2017), where the local solver IPOPT was shown to find solutions that are practically indistinguistable from those obtained using COUENNE and Lasserre hierarches for pipe flow problems. The equations used to model pipe flow are closely related to the Saint-Venant equations.

Numerical benchmarks. In Chapter 6, the solution quality and run time of IPOPT is compared with the solution quality and run time of *RGA* (Vermuyten et al. 2018), a type of genetic algorithm, for a suite of hydrodynamic benchmark problems. It demonstrates the excellent solution quality obtained using local methods and thereby provides further numerical evidence for the invex optimization theory of Chapter 5.

Practical applications. In Chapter 2, linear mass balance models are applied in practice for water allocation in the Netherlands at national scale. The resulting tool is used by *Rijkswaterstaat*, the executive branch of the Dutch Ministry of Infrastructure and Water Management, to create and analyze strategies for the mitigation of the effects of drought. The approach is, furthermore, being used at Deltares to construct a more detailed optimization model for the *Nationaal Hydrologisch Instrumentarium* (national hydrological instrument), including not only the primary waterways but also lesser water bodies.

In Chapter 7, the non-convex local method is applied in practice for the MPC of the primary canal system of the water authority *Hoogheemraadschap van Rijnland* in the Netherlands. The area

under its authority lies between the North Sea, Amsterdam, and The Hague, and includes the city of Leiden as well as Amsterdam Schiphol Airport. To the author's knowledge, this is the first system of its kind being operational anywhere in the world. The numerical performance of the system and the quality of the solutions is such that the operators at Rijnland leave the water level control of the system in the hands of the algorithm for extended periods of time. Similar applications, in various stages of operationalization, have been developed for the water authority *Noorderzijlvest* (by Deltares), for the water authority *Hoogheemraadschap de Stichtse Rijnlanden* (by KISTERS), and for a major Canadian hydropower utility (by KISTERS), with several more applications in the planning phase, including for the Ganges basin in India (by KISTERS).

1.8 Overview of chapters

The rest of this thesis consists of six chapters. Here, the contributions of each chapter are given.

In Chapter 2, the primary waterways in the Netherlands are modelled using a linear mass balance formulation, *i.e.*, an integrator-delay model with zero delay. Together with a multi-objective goal programming formulation, this results in a cascade of convex optimization problems. The resulting tool is used by *Rijkswaterstaat*, the executive branch of the Dutch Ministry of Infrastructure and Water Management.

In Chapter 3, the linear mass balance formulation is extended to cover a convex approximation of the pump equation (1.6). The relationship between flow, hydraulic head, and power consumption of a pump is roughly bilinear. A suitable convex approximation allows energy and cost minimization to be posed as convex prob-

lems.

In Chapter 4, the machinery of Chapter 3 is extended to cover adjustable weirs, *i.e.*, equation (1.5).

Chapter 5 provides a general framework for analyzing the global optimality properties of solutions to non-convex discrete-time optimal control problems. This enables a principled approach towards using the full Saint-Venant equations, as well as bilinear pump and turbine equations, in optimization problems.

Chapter 6 compares the solution quality and run time of a local method with the solution quality and run time of a type of genetic algorithm, for a suite of hydraulic benchmark problems that satisfy the regularity conditions of Chapter 5. It demonstrates the excellent solution quality obtained using local methods for such problems.

Chapter 7 describes a real-life application of optimization subject to the Saint-Venant equations. This application is used on a daily basis by the *Hoogheemraadschap van Rijnland*, *i.e.*, the water authority of Rijnland, for the control of the primary canal system in one of the economically most important regions of the Netherlands.

1.8 Disclosure

Chapterwise, this thesis is based on the following four research papers:

Chapter 2 P.J.A. Gijsbers, J.H. Baayen and G.J. ter Maat. Quick Scan Tool for water allocation in the Netherlands. *Environmental Software Systems: Computer Science for Environmental Protection*, Springer, 2017.

Chapter 3 K. Horváth, B. van Esch, D.J. Vreeken, I. Pothof and J.H. Baayen. Convex modeling of pumps in order to optimize their energy use. *Water Resources Research*, volume 55, issue 3, pp. 2432–2445, 2019.

Chapter 4 K. Horváth, B. van Esch, I. Pothof, D.J. Vreeken, J. Talsma and J.H. Baayen. Closed-loop model predictive control with mixed-integer optimization of a river reach with weirs. *IFAC-PapersOnLine*, volume 52, issue 23, pp. 81–87, 2019.

Chapter 5 J.H. Baayen and K. Postek. Hidden invariant convexity for global and conic-intersection optimality guarantees in discrete-time optimal control. Accepted for publication in *Journal of Global Optimization*, 2021.

Chapter 6 contains material from: J.H. Baayen, D. J. Vreeken and P. Archambeau. Optimization methods for hydraulic systems. *6th International symposium on the hydrological modelling of the Meuse basin*, 2019.

Chapter 7 contains Section 8.2 from: J.H. Baayen, B. Becker, K.-J. van Heeringen, I. Miltenburg, T. Piovesan, J. Rauw, M. den Toom and J. VanderWees. An overview of continuation methods for nonlinear model predictive control of water systems. *IFAC-PapersOnLine*, volume 52, issue 23, pp. 73–80, 2019.

Each chapter contains contributions from all of its respective authors. All numerical experiments were executed by me, except for the RGA runs of Chapter 5, and except the case studies of Chapters 3 and 4. These two chapters were revised substantially for this dissertation. The applications of Chapters 2 and 7 were a joint effort of all respective authors. The numerical experiments of Chapters 2, 3, and 4, as well as the application of Chapter 7, were implemented using RTC-Tools 2 (Baayen et al. 2018), a software package of which I was the architect and lead developer.

Part I

Convex approximation of flow network problems

CHAPTER 2

Quick Scan Tool for water allocation in the Netherlands

2.1 Introduction

2.1.1 Freshwater availability in the Netherlands

The Netherlands is rich in water. In the current situation there is hardly any water scarcity, *i.e.*, a situation in which usual consumption rates exceed the average water availability. However, the country has to cope with droughts, the natural phenomena in which there is temporarily decrease in fresh water availability, as occurred during extreme dry historical years 1976 and 2003.

The majority of the water system allows controlled redirection of water where most regions can be supplied from the national water system during dry periods, using the Rhine River and Meuse River as the main sources. After the Rhine and Meuse enters the Netherlands the water is distributed over the branches Waal, Nederrijn, and IJssel by means of a weir by Driel. In general, 2/3 of the inflow goes to the Waal, and 1/3 to the Nederrijn and IJssel. The IJssel supplies the IJsselmeer and Markermeer lakes with fresh water. From the rivers and lakes, water is distributed to other parts of the country through an extensive network of ditches

and canals (Haasnoot et al. 2014).

Water is used for controlling levels, for controlling water quality by means of flushing, and for actual extractions (*e.g.*, irrigation and drinking water). In the Netherlands there is no absolute shortage of water but a "problem" of the right quality at the right time at the right place. Specific phenomena for the Netherlands as a low-lying country is the risk of salt water intrusion in the western part of the Netherlands; during low flow of rivers sea water enters the main water ways due to the lack of driving forces pressurizing the water supply of the western part of the Netherlands. As described above, in the north water can be stored in the IJsselmeer and Markermeer, a man-made lake (total 2000 ha), that was created in 1932 by building a large dike (the Afsluitdijk) and is fed by the River IJssel. The lake supplies the northern provinces during summertime. Groundwater is the main water source in the elevated areas in the south and east, since these grounds cannot be reached by re-routed surface water from the rivers. Finally the islands and peninsulas in the south west have to deal with both saltwater intrusion and limited fresh water supply options due to their surrounding by salt water bodies.

Policy arrangements are in place and coordinated by the National Coordination Committee for Drought Conditions (LCW) to deal with reduced water availability conditions. Leading principle is the so-called *Verdringingsreeks*, a priority lists which puts infrastructure and nature integrity preservation purposes such as water level and water quality control (flushing) above extractions for drinking and industry water above irrigation for agriculture.

2.1.2 The Adaptation Challenge

Climate change scenarios provided by the Royal Netherlands Meteorological Institute (Hurk et al. 2014) indicate that the Netherlands should expect both an increase in the number and extent of high flow events as well as low flow periods. It is expected that global climate change demands that various measures will be taken to guarantee the control of water levels and a supply of freshwater for the long term (at least until 2100). Socio-economic developments could raise water demands even further, beyond just climate change impacts. A quantitative assessment of both problems and solutions in collaboration with stakeholders should indicate if this expectation could come true and if yes, underpin to what extent one should expect this to happen.

In the previous phase of the policy process, the Dutch Delta Programme Phase 1, which ended in 2015, an adaptive pathways plan was presented to cope with droughts and water scarcity on the short term (until 2021), midterm (2050) and long term (2100) (Haasnoot et al. 2012). Funds were allocated for measures and on different levels: for the main water system, regional water system and for water users, that are agreed upon as implementation of the preferred pathway. For the short term it was also agreed upon to make the current system more flexible and robust by "smart" operational water management.

2.1.3 The Need to Support Screening of Alternative Strategies

In 2008-2009 various Dutch institutes conducting water management research started a collaboration to jointly develop a National Hydrological Model (De Lange et al. 2014). This National Hydrological Model (NHM) is a detailed integrated surface-groundwater

system of the Netherlands, combining a 250×250 m grid model for groundwater (Modflow) and unsaturated zone (Metaswap) with a surface water balance and water allocation model consisting of 8800 network elements. This water allocation model uses a heuristic, rule-based approach to allocate water to various prioritized purposes where the uses with high priority receive water before uses with lower priority. The NHM is applied for policy analysis and for operational forecasting.

For the operational forecasting application, the NHM is encapsulated in a Delft-FEWS based operational forecasting system call RWsOS-Waterbeheer (Weerts et al. 2011). Rijkswaterstaat, the Dutch national authority responsible for the national waterways and water bodies, uses RWsOS-Waterbeheer on a daily basis to produce a real-time forecast with a 10 day horizon using the NHM. This operational system provides useful information to the National Coordination Committee for Drought Conditions to analyze the current water availability situation in the Netherlands. The Committee has the authority to change the water allocation at a national level if the drought situation is sufficient severe. For this purpose, it wants to be able to conduct what-if analysis runs such that trade-offs between different regions and sectors can be assessed when allocation patterns are changed. Since the runtime of the detailed NHM is substantial, taking hours to days, a need arose for a so-called Quick Scan Tool to accommodate this screening purpose. This operational tool will be referenced as the LCW-Quick Scan Tool.

In the policy analysis domain the NHM is encapsulated in a larger modeling system called the National Water Model. The model is used for detailed policy analysis under future climate scenarios and socio-economic developments. Simulations, conducted for a 30 - 100 year time series targeted at a mid-term 21st century

outlook (2050), take days to weeks. This makes it hard to use the model for an initial screening of interventions to address the issues that arise in terms of water supply and saline intrusion within a dynamic multi-stakeholder policy process. Also for these studies a need arose for a Quick Scan Tool that could be applied to analyze potential interventions for current or future bottlenecks in the water supply system. This policy analysis tool will be referenced as the PA-Quick Scan Tool. Once interesting interventions have been identified, they could be implemented in the NHM of the Netherlands to conduct a detailed analysis.

2.1.4 Quick Scan Tool Requirements

While the two tools have different end users, they have also many similarities. The end users for the LCW-Quick Scan Tool are civil servants, namely the core members of the LCW itself. These people intend to use the tool as a preparation to the Committee meetings to investigate alternative water allocation strategies when hydrological conditions are becoming dry. Their tool needs to be based on the most recent datasets provided by RWsOS-Waterbeheer. The LCW-Quick Scan Tool must be easy and quick to use and provide insight in the current situation as well as the regional trade-off of water balance effects of alternative allocation strategies. The end users for the PA-Quick Scan Tool are Deltares experts conducting the policy analysis for different climate and socio-economic development scenarios. The input datasets for the PA-Quick Scan Tool will be 30 year time series of water demands and river discharge, provided by the National Water Model.

Both applications require a water balance model that can cope with priority-based water management rules for extractions and management of lake levels. The challenge for modeling the national Dutch water system, characterized by a high-density net-

work of waterways, is to design a model schematization that represents the national water system with the main water delivery routes and storages appropriately while regional detail should be neglected when the issues at stake do not ask for those details. This network model has to accommodate regional trade-off analysis as well as analysis of the issues at stake, while potential interventions should transparently be facilitated in the parameterization of the model. Possible interventions can vary from changing requests (water demands for extraction and in-stream uses) to manipulation of the operating rules for the storages and modification of maximum intake capacities. To enable quick turnaround times in the analysis, all above interventions should be facilitated via a Graphical User Interface. Changing the order of water use priorities was left out of scope as reduction of water demands could be applied to accommodate analysis of the same intervention in water shortage conditions.

2.2 Methods and techniques

The model underlying the Quick Scan Tool is composed of a coarse network of the Dutch water system including the major water storages, water distribution points and delivery routes to the various uses. Requests for water abstractions (agriculture, industry, drinking water, regional water systems) and instream flows for flushing are prioritized and assigned to the nodes and links in the network.

In the present section we discuss the methods and techniques used to develop a model for solving our water allocation problem. In particular, we will be using mathematical optimization to implement priority-based water allocation.

The central design tenet of our tool is separation of concerns, *i.e.*,

implementation of conceptually disjunct functionalities in separate modules. In our case, the physical system model is kept separate from the specification of the water allocation goals. In the subsequent sections, we will discuss the methodologies underpinning the implementation of these two modules.

2.2.1 Modeling the water system

The Dutch water system may be viewed as a network composed of elementary objects, such as:

- storage nodes,
- channel reaches,
- weirs and pumping stations.

Objects of the same type share the same parameterized equations governing their dynamics. Storage nodes, for instance, are governed by the mass balance equation:

$$\frac{dV}{dt} = Q_I - Q_O,$$

with storage volume V, inflow Q_I and outflow Q_O. This is a differential equation.

Instantaneous routing in a channel reach is governed by algebraic equations of type

$$Q_{down} = Q_{up},$$

with upstream inflow Q_{up} and downstream outflow Q_{down}.

Collecting the equations for all the network elements, results in a system of *differential-algebraic equations* (DAE). For the Quick Scan Tool a need was identified for a system to describe classes

of model elements using DAE, and to combine instances of these classes into a network model. A modeling language that allows this is *Modelica* (Elmqvist 1997). The Modelica objects used for the Quick Scan Tool are reaches with instantaneous routing with an extra term for a lateral flux (discharge/extraction), connection nodes for network confluences and diversions, and storage nodes.

2.2.2 Optimization with prioritized goals

The priority ordering of the control goals of the Dutch water system (the *Verdringingsreeks*) leads us to consider sequential optimization of the prioritized goals in order. In operations research, this technique is known as *lexicographic goal programming* (LGP) (Collette and Siarry 2003, Eschenbach et al. 2001). The idea of LGP is to optimize the convex goal[1] functions f_i in a given order, prioritizing earlier goals over later goals. The goals are ordered by assigning each a non-negative integer priority value p_i. The goals are then solved in their priority order. Following the optimization of a goal function f_i, its attainment level is fixed and added as a constraint to the optimization problem. The optimization of all following goals, in this way, will not worsen the attainment of any preceding goal. At each stage of LGP, optimization takes place within the degrees of freedom left open by the fixation of the attainment levels of the previous goals.

Solving ordered goals. Application of LGP to a multi-objective optimization problem results in a series of optimization problems. Let k be the priority level under consideration, and let the overall problem be constrained by the inequality $g(x) \leq 0$ with affine

[1]Within this chapter, the terms *goal* and *objective* are used interchangeably.

vector function g. The k'th optimization problem is then

$$\begin{aligned} \min_x \ & f_k(x) \\ \text{s.t. } & g(x) \leq 0 \\ & f_i(x) \leq \varepsilon_i \quad \forall i < k, \end{aligned}$$

with the attainment level of the ith goal

$$\varepsilon_i := f_i(x_{opt,i})$$

and $x_{opt,i}$ being the optimal solution of the ith optimization problem.

LGP has been applied to decision support for the short-term operation of hydropower resources (Eschenbach et al. 2001, e.g.) and to surface water allocation (McGregor and Dent 1993, e.g).

Inequality goals. In water systems, one often encounters the need to keep variables within a desired range. A channel reach is a typical case, where one aims to keep the water level within desired lower and upper bounds. It may not always be possible to keep the water level within the desired range, as in case of drought or flooding. Inequality, or range, goals are therefore an important ingredient in a multi-objective optimization framework.

Let h_i be a goal function defined for a single time step (*e.g.* a water level at a certain point in space in time) and let $[m_i, M_i]$ be its desired range, with $m_i \in \mathbb{R} \cup \{-\infty\}$, $M_i \in \mathbb{R} \cup \{+\infty\}$ and $m_i \leq M_i$. Let k be the priority order level under consideration, and let the

overall problem be constrained by the equation $g(x) \leq 0$. The k'th optimization problem is then

$$
\begin{aligned}
\min_{x,\delta_k} \ & \|\delta_k\|_p^p \\
\text{s.t. } & g(x) \leq 0, \\
& h_k(x(t_j)) \geq m_k + \delta_k(t_j)(\underline{h}_k - m_k) && \forall j = 1,\dots,T, \\
& h_k(x(t_j)) \leq M_k + \delta_k(t_j)(\overline{h}_k - M_k) && \forall j = 1,\dots,T, \\
& \delta_k(t_j) \geq 0 && \forall j = 1,\dots,T, \\
& \delta_k(t_j) \leq 1 && \forall j = 1,\dots,T, \\
& h_i(x(t_j)) \geq m_i + \delta_i(t_j)(\underline{h}_i - m_i) && \forall j = 1,\dots,T, \\
& && \forall i = 1,\dots,k-1, \\
& h_i(x(t_j)) \leq M_i + \delta_i(t_j)(\overline{h}_i - M_i) && \forall j = 1,\dots,T, \\
& && \forall i = 1,\dots,k-1,
\end{aligned}
$$

with vector violation variable δ_k and goal function enclosure (Moore and Bierbaum 1979) $[\underline{h}_k, \overline{h}_k]$ such that $\underline{h}_k \leq h_k(x(t_j)) \leq \overline{h}_k$ for all feasible x and all time steps t_j, $j = 1,\dots,T$, where T denotes the final time step along the prediction horizon. The order $p \geq 1$ denotes the norm under consideration. One would select $p = 1$ for linear penalization, or $p = 2$ to penalize large violations disproportionately more than small ones. Since the variables $\delta_k(t_j)$, $j = 1,\dots,T$, are non-negative, the optimization problem is convex for all $p \geq 1$. The concept of violation variables is illustrated in Figure 2.1 (left). For any $j = 1,\dots,T$, the goal function merely lies within its enclosure when $\delta_k(t_j) = 1$, whereas the goal is fully satisfied when $\delta_k(t_j) = 0$.

In addition, for every $i < k$ and every $j = 1,\dots,T$, the following constraint is added to fix the goal attainment level:

$$m_i + \delta_i(t_j)(\underline{h}_i - m_i) \leq h_i(x(t_j)) \leq M_i + \delta_i(t_j)(\overline{h}_i - M_i).$$

For all $i < k$, the violation values $\delta_i(t_j)$, $j = 1, \ldots, T$, are taken constant.

The effect of an inequality goal is best described as a *soft constraint*. First, the optimizer will try to find a state trajectory that lies within the desired range. All trajectories that lie within the range incur no penalty cost and are therefore equally preferable. If it is not possible to find a trajectory that fully lies within the desired range, the optimizer will select a trajectory that lies outside of it as little as possible. The desired range, relaxed just enough to accommodate the actual trajectory, is taken as a standard (hard) constraint for subsequent goals. This idea is illustrated in Figure 2.1 (right).

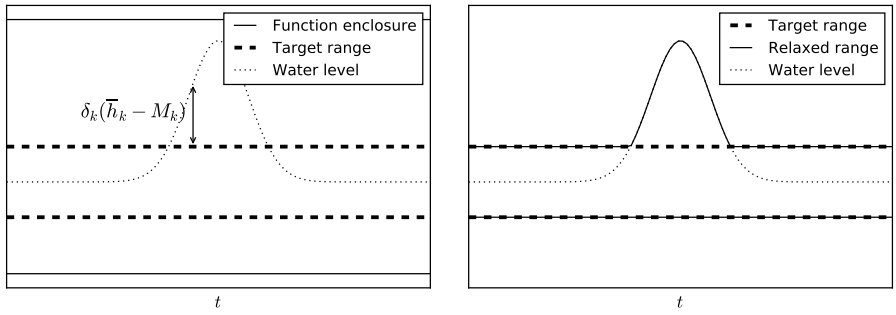


Figure 2.1 – Variable violation concept (left) and relaxation of state beyond the target range (right).

LGP with inequality goals has been applied to decision support for the short-term operation of hydropower resources (Eschenbach et al. 2001).

Multiple goals per priority level. In the Quick Scan Tool, every priority level i may come with multiple goal functions $\{f_i^k\}_k$, each covering a different element of the model. These goals are assumed to be equally important, hence allowing the respective

goal functions to be summed into a single objective function f_i:

$$f_i = \sum_k f_i^k.$$

Optimization solver. The interior point solver IPOPT (Wächter and Biegler 2006) is used to solve the convex optimization problems resulting from the lexicographic optimization procedure. IPOPT solves convex problems with nonlinear objective, which is required for goals with order $p > 1$.

Initialization. The optimization problem for every subsequent priority level is initialized with the solution from the previous level. New violation variables at priority k are initialized as $\delta_k(t_j) = 1,\ j = 1, \ldots, T$, to ensure that the new optimization problem is feasible.

Software suite. The techniques covered in the preceding sections are available as standard components in the environmental flow optimization software suite RTC-Tools 2.0 (Baayen et al. 2018), which supports Modelica model formulations. RTC-Tools is available under a dual-licensing scheme. The open source version is available online under the terms of the GNU General Public License version 3.

2.3 The Quick Scan Tool application

The Quick Scan Tool application is a combination of two software products. Delft-FEWS provides the Graphical User Interface, the database and general data processing capabilities. The model component uses an internal model based on the RTC-Tools 2 model engine (Baayen et al. 2018).

2.3.1 The model schematization

The Quick Scan Tool holds an internal network model for the water balance. The model, built in Modelica, is composed of model element of type branches, connection nodes, demand nodes, storages as well as boundary nodes (inflow and terminal). Each Modelica element holds its own water balance where flow enters via the "in"-port and leaves via the "out"-port. Lateral flows (extractions of discharges) can be applied to branches, demand nodes and storages.

The final network model schematization (Figure 2.2) is the result of five collaborative design sessions with the end users. The challenge was to design a network which is as simple as possible while providing an appropriate representation of the main water system including the main water delivery routes. The network model should accommodate analysis of the issues at hand while potential interventions should be facilitated transparently in the parameterization. The result is a schematization where areas in the north-east and in the south-east are grossly simplified compared to the actual water system as there are only few relevant water inlets. More network detail has been introduced in the west as many water delivery routes need to be analyzed in relation to saline intrusion of the most westward inlets. Capacities and management rule have been derived from the NHM.

The model introduces 374 variables and 226 linear constraints per time step considered. A further 369 violation variables and linear constraints are introduced by LGP at the last priority level, also per time step considered. For 40 time steps, this results in approximately $30,000$ variables and $24,000$ constraints.

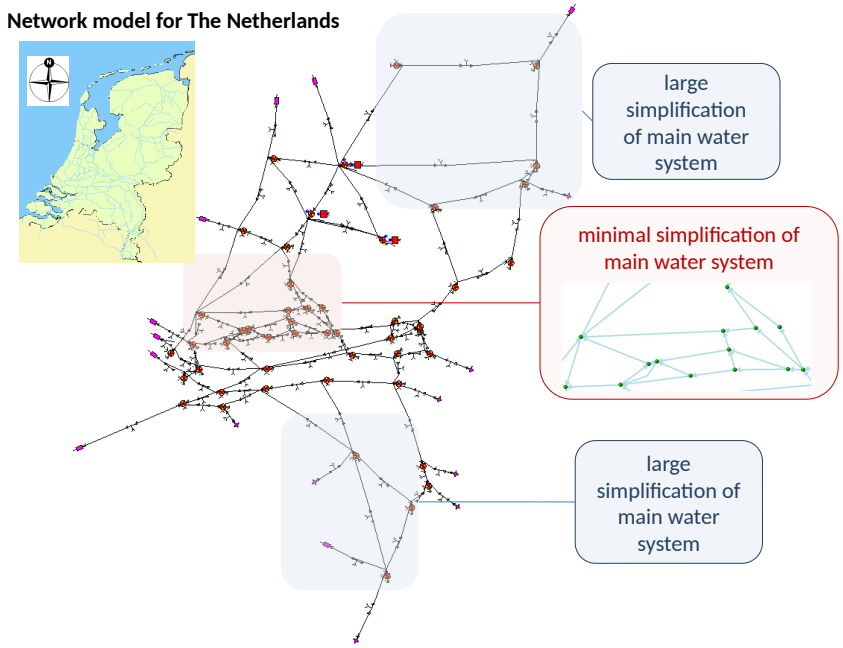


Figure 2.2 – QST network schematization of the main water delivery system in the Netherlands.

2.3.2 The work and dataflow of the application

The data flow of the Quick Scan Tool is composed of the following steps:

1. Acquire the input data sets (requests for extractions and flushing), river discharges and lake levels) from the National Hydrological Model.
2. Conduct a spatial assignment, using a coupling table, of all requests to a node or branch of the coarse network model.
3. Transform request by water use function to a request by priority.

4. Allocate the available water resources using the LGP method. The result is a water distribution over the network, with per element the inflow, the outflow and the lateral flux achieved. Since this lateral flux is the total flux for all extracting (and discharging) water uses functions, this flux needs to be split.

5. Split the lateral flux into the portions allocated to each water use.

6. Aggregate the results by area for presentation purposes.

7. Compute a delivery rate percentage, broken up by water use, for presentation purposes.

Table 2.1 illustrates the order of priorities that is adopted in the PA-Quick Scan Tool.

Table 2.1 – Ordering of goals as applied in the PA-Quick Scan Tool.

Priority	**Model variable**	**Lower bound**	**Upper bound**
1	Lateral	Natural loss/contribution	Maximum extraction
2	Inflow	Min. capacity (physical)	Max. capacity (physical)
3	Level	Min. storage level (physical)	Max. storage level (physical)
4	Outflow	Request water level preservation	
5	Level	Min. storage (low priority)	Max. storage (low priority)
6	Lateral		Request Utilities
7	Outflow	Request navigation locks	
8	Level	Min. storage (middle priority)	Max. storage (middle priority)
9	Lateral		Request rural water system
10	Level	Min. storage (high priority)	Max. storage (high priority)
11	Outflow	Request network flushing	
12	Level	Target storage level	Target storage level
13	Outflow		

The first three priorities are critical, *i.e.*, similar to standard (hard) constraints, as they intend to force the solution within the physical bounds of the water system. The remaining goals are water uses with their order prescribed by law and policies. As can be

noted, the operating rules for the storages are divided in multiple goals such that lake levels can drop to meet high priority water demands. Target storage levels should be attained if sufficient water is available within the system. The goal with the last priority intends to minimize the outflow in a selected set of branches such that the water remains in the main rivers and only enters the water inlets to meet local water demands. All canal outlets that do not conduct a water delivery function to downstream uses should be included in this selection.

For in-stream flow requests (*e.g.*, flushing and navigation locks), each new goal is identified by taking the maximum of all in-stream water use flow requests up till the priority at stake. For the lateral fluxes the LGP approach intends to squeeze the solution space with each priority such that the end result is at the desired request if the system is not under water shortage. The equations in Figure 2.3 indicate that the requested lateral fluxes for different water uses need to be stacked to obtain a series of goals that squeeze the solution space in the ordering of goals.

Once the LGP has completed its computation, the resulting flows need to be split according to the different water uses. For any in-stream flow request the request is fulfilled if the realized flow is larger than the request. For lateral flux requests, the delivery per water use needs to be based on the "peeling off" of the realized lateral flux by order of priority and allocating the remaining flux to a specific water use up to the request.

The total procedure, including the optimization steps, takes approximately 10s on a modern notebook.

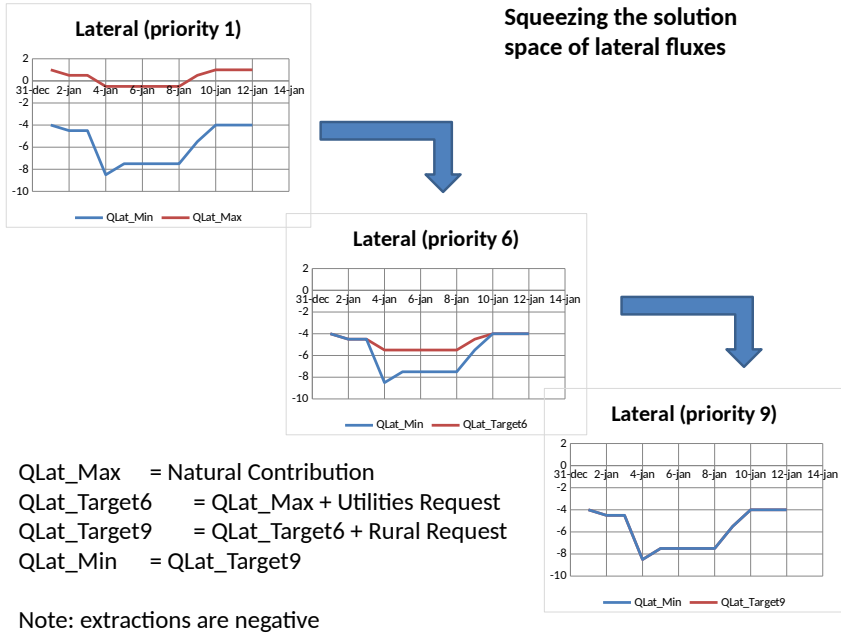


Figure 2.3 – Solution space squeezing and associated goal stacking for lateral fluxes.

2.4 Validation of model results

The QST-model has been validated by comparison of the optimized water allocation against the water allocation result computed in the detailed NHM. Some differences are to be expected, given that the QST uses a lower resolution model and optimization to allocate water. The NHM, on the other hand, uses a high-resolution model and rule-based heuristics to allocate water. Important items checked are the distribution of the main river inflows, the behavior of the lakes, the flushing on the outlets and the main inlets for the rural area in the west. Figure 2.4 illustrates the water shortage at a downstream reach (*Nieuwe Waterweg*) for the flushing goal at the driest moment in the 30 year historical

series, flushing being the goal with the lowest priority. In general the QST model results in fewer shortages, which is likely due to the use of optimization. During the driest period, the difference with the NHM model result is $2-6\%$ at this location, while most other network elements show hardly any difference. Also lake levels follow the same pattern between the two models if the QST is to lower the volume in order to meet in-stream flow requests. The accuracy is considered acceptable for screening purposes.

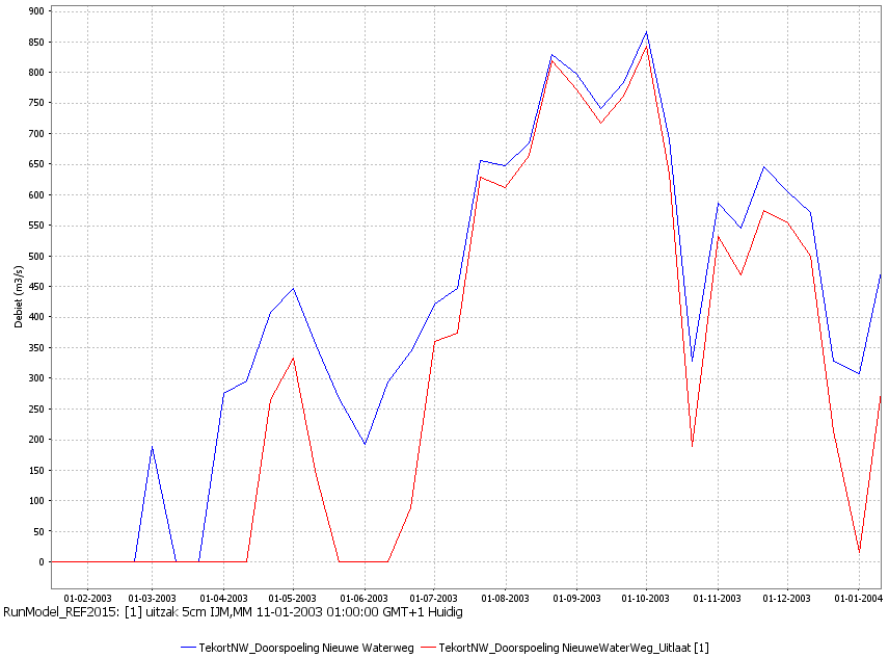


Figure 2.4 – Comparison of shortage at sea outlet between NHM (blue) and QST-model (red). The y-axis represents discharge (*debiet* in Dutch).

2.5 Discussion and conclusions

Within the Netherlands, a need arose at the national level to supplement the current high-resolution National Hydrological Model with a water allocation model to enable the screening of national water allocation decisions for their regional trade-offs. This chapter illustrates how a so-called Quick Scan Tool has been developed for this purpose. The tool is composed of a water allocation model engine and a software component that offers a front end as well as pre- and post-processing capabilities. The water allocation model is built with the RTC-Tools 2 software framework using a lexicographic goal programming approach to priority based water allocation rules. In the pilot version Excel was chosen as the front end offering application-specific pre- and postprocessing capabilities. Data processing required extensive handling of lookup tables while conducting the necessary aggregations from detailed source data to coarse model input to aggregated output for presentation on areal scale. The prototype application showed that RTC-Tools 2 was meeting the model needs, while Excel was not the appropriate data processing platform. In the final version, the Delft-FEWS software framework was chosen as a data processing and visualization platform. This platform offered much more insight when debugging in- and output data flows.

The resulting Quick Scan Tool has been validated against the detailed National Hydrological Model to confirm that the tool outcomes are sufficiently similar to this accepted model to support the policy process. The visualization capabilities Delft-FEWS, with its map based flow animations and graphs, were very beneficial while discussing model setup and validation results with the water management experts of Rijkswaterstaat, the governmental body who initiated the development of the tool.

The tool has been in continuous use for policy analysis from 2017

up to and including the time of this writing (2021).

In the near term, functional extensions are foreseen using meta-relations to transform reduced river flows into impacts on salinity rates and navigation depths.

RTC-Tools 2 and Delft-FEWS are software platforms which initially were developed for near real-time water system operations. Based on this Quick Scan Tool, it can be concluded that the flexibility and customization capabilities of these platforms also allow development of decision support tools for strategic planning processes.

CHAPTER 3

Convex modeling of pumps in order to optimize their energy use

3.1 Introduction

Freshwater is a basic resource. However, freshwater is often distributed unevenly. To correct this distribution and prevent the problems originating from it, operational water management ensures the right amount of water at the right place and time. The distribution of water is carried out by hydraulic structures, such as weirs and pumps. The decision about the operation of these structures determines the distribution of water. These principles hold for different water systems, such as irrigation and drainage systems or water distribution networks. To make operational decisions, mathematical optimization is used to enable water managers to take into account future water demands and weather forecasts at the time of decision making. For example, for a drainage canal the task of the optimization is to determine the pump operation such that the water level in the canal stays in the prescribed bounds while the nearby areas are drained into that canal. In doing so, disturbances, such as rainfall, should be taken into account. Mathematical optimization of water systems requires a suitable model for every system component. This work focuses on

the derivation of approximate pump models that are appropriate for energy consumption minimization purposes. The approach is demonstrated on a drainage system, but it can be applied to other systems, such as ground water pumping or to water distribution networks.

Modeling and optimization of pump operation can be found in the literature mainly in the context of water distribution networks. Mala-Jetmarova et al. (2017) provides an overview of this topic. The reason why optimization is studied mainly in connection with water distribution systems is that the operational expenditures of such a network are strongly related to pumping costs (Ormsbee and Lansey 1994). Pumps fall into one of two categories: constant-speed pumps, whose speed does not vary, and variable-speed pumps, whose speed varies depending on the discharge and the head.

Constant-speed pumps are often used in water distribution networks. For such pumps, the optimization algorithm only has to determine when the pump should be switched on and off. This is a mixed-integer optimization problem, which may furthermore have a non-convex continuous structure, whence it can be very time-consuming to find a global optimum. For this reason, pumps in water distribution networks are often optimized by faster algorithms providing only suboptimal solutions. In Marchi et al. (2017), for instance, a genetic algorithm is used combined with an accurate dynamic pump model. Heuristics can be used to increase the solution speed (Fatemi et al. 2017). They also, however, result in suboptimal solutions in general.

There are few methods in the literature that can provide a global optimum within acceptable time-frames. Cembrano et al. (2000) avoids using mixed-integer optimization by calculating the volume to be pumped in the formulation (instead of the on-off state

of the pump) resulting in a continuous problem, and then the pump-schedule is obtained via post-processing. However, in this case the number of pump-switching times must be known a priori, which is not the case most of the time. Lansey and Awumah (1994) develop simplified hydraulic models for each possible pump combination, but the applicability of the model is limited by the number of pumps in the system. Dorini et al. (2012) describe convex modelling and optimization of pumps, supposing they are always on. Another way to reach the optimum without having to solve a mixed-integer problem is to use switching time as the optimization variable to determine the start and length of the pump operation (Dekens et al. 2014, Price and Ostfeld 2014). This approach provides a mathematical solution to the problem that leads to a global optimum; however, it becomes very complex when it is combined with other elements of the water system, because the objective of the other elements might not be expressible using switching times as optimization variables. Moreover, similarly to the previous case, the number of pump switches should be set a priori. Although the last two methods are able to reach a global optimum (even having their own limitations), they can only be applied to constant-speed pumps. They are not suitable to minimize the energy use of variable-speed pumps.

The optimization of variable-speed pumps is a challenging task. Similar to constant-speed pumps, variable-speed pumps can be seen as discrete systems - as they have on and off positions - and hence their optimization is a mixed-integer problem. As opposed to constant-speed pumps, not only the time of operation should be determined, but also the shaft speed. Due to this complexity several authors have used heuristic methods to solve the optimization problem. Commonly used heuristic optimization methods can only achieve a suboptimal solution, for example ant-colony optimization (Hashemi et al. 2014), stochastic optimization

techniques such as Particle Swarm Optimization (Wegley et al. 2000), or genetic algorithm (Olszewski 2016). In Bagloee et al. (2018), Candelieri et al. (2018), Marchi et al. (2016), Oikonomou et al. (2018) machine learning techniques are combined with optimization methods. Other applications include pumped-storage hydropower plants (Alizadeh-Mousavi and Nick 2016, Schmidt et al. 2017), and the use of pumps as turbines (Fecarotta et al. 2016). Malrait et al. (2017) minimize the motor losses of the pump for a given hydraulic operating point (*i.e.*, pump speed), thus it does not optimize the whole hydraulic system. Another way to tackle the problem of approximating an optimum of the discrete system is using the solution of the continuous system: Ulanicki and Kennedy (1994) solve a non-convex continuous problem and rounds the solution of the continuous optimization to integers. This method also leads to suboptimal solutions.

None of the above mentioned methods for variable-speed pumps are able to reach a global optimum due to the use of heuristic optimization methods. One algorithm that does find global optima is presented in (Menke et al. 2015, 2016). The authors use simplified modeling of pumps which enables the use of mixed-integer convex optimization to reach a global optimum. Moreover, they show that linear approximations outperform nonlinear approximations in computation time without significant loss of accuracy. They use convex approximations of the pump curves combined with a relatively high (5%) optimality gap in the branch and bound algorithm to be able to find the optimal solution within affordable computation time. The power consumption of the fixed-speed pump is assumed to be constant, and for variable-speed pumps depends only on the discharge. However, the actual power consumption of the pump highly depends on discharge and head. Therefore, this method introduces a large error in power approximation, thus in the estimated energy consumption.

In this work, we present a convex approximation of the pump scheduling problem. Locally optimal solutions to this approximation (in the continuous sense) are globally optimal and minimize energy use by taking into account its dependence on head and flow conditions.

3.2 Methodology

In this section the convex approximation of the pump scheduling problem is described. First, the operation of variable-speed pumps is recalled. Next, the case is described when the pump is always switched on and formulated as a continuous convex optimization problem. Then, the possibility to switch off the pump is included. This requires mixed-integer convex modeling of the pump optimization problem, which accounts for the main contribution of this chapter. Finally, the formulation is extended to constant-speed pumps.

3.2.1 Description of variable-speed pumps

In this section the operation of variable-speed pumps is described. A pump is a dynamic device that increases the pressure of a liquid by transferring the mechanical energy of the rotating impeller to the liquid (Wright and Gerhart 2009). See Figure 3.1. The pressure head generated by the pump at a certain shaft speed depends on the discharge. This dependency is presented graphically by the so-called $(Q, \Delta H)$-curve of the pump. Figure 3.2 presents an example of such a curve for a large axial flow pump used for polder dewatering, running at 356 rpm.

Apart from the $(Q, \Delta H)$-curves, characteristic curves for power and efficiency are also included in the figure and are seen to depend on discharge as well. The manometric head, ΔH_{man}, generated

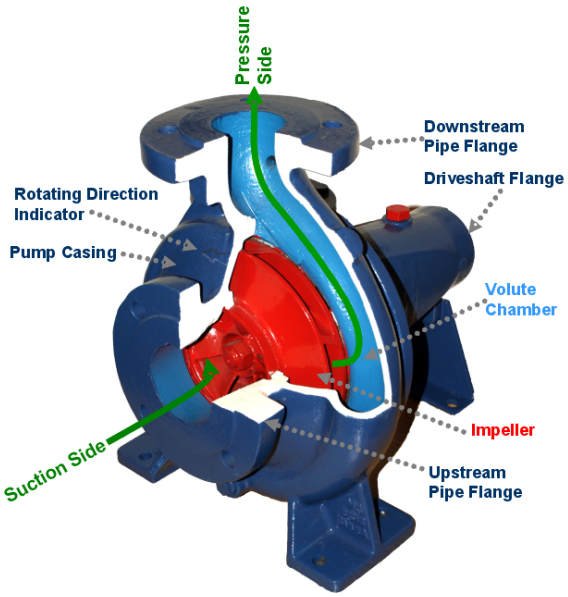


Figure 3.1 – Centrifugal pump. Source: Wikipedia.

by the pump is partially used to compensate the pressure losses in the pumping station and partially to overcome the total, static head ΔH, *i.e.*, the difference in water levels between the two sides of the pumping station.

For every shaft speed, there is a corresponding $(Q, \Delta H)$-curve. Thus the relationship between discharge, head and shaft speed is unique, any two determine the third. This relationship is described by the affinity laws. The performance of an axial flow pump is given in Figure 3.3 for a range of different shaft speeds, showing how the pump can be used for a range of discharges and heads. However, in practice the operating range is limited by several factors, like minimum discharge Q_{min} to prevent unstable operation, cavitation if available net positive suction head (NPSH) is not sufficient, and maximum power P_{max} of the motor. All these curves together define an area where the pump can be operated:

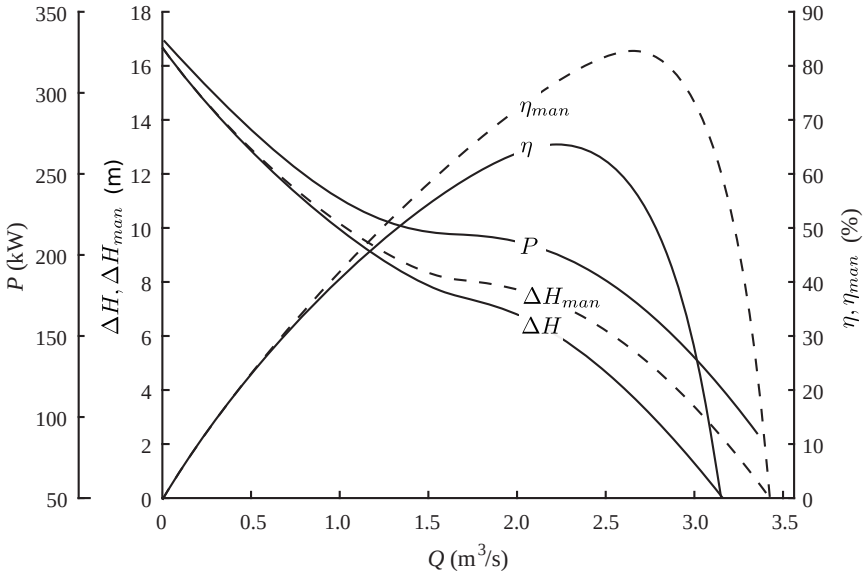


Figure 3.2 – Pump characteristic curves for an axial flow pump with impeller diameter 1040 mm running at 356 rpm. Shown are characteristics for manometric head ΔH_{man}, total or static head ΔH, power P, manometric efficiency η_{man}, and total efficiency η.

this is called working area in the following. In case of constant-speed pumps there is only one $(Q, \Delta H)$-curve, and the working area is reduced to a section of the curve.

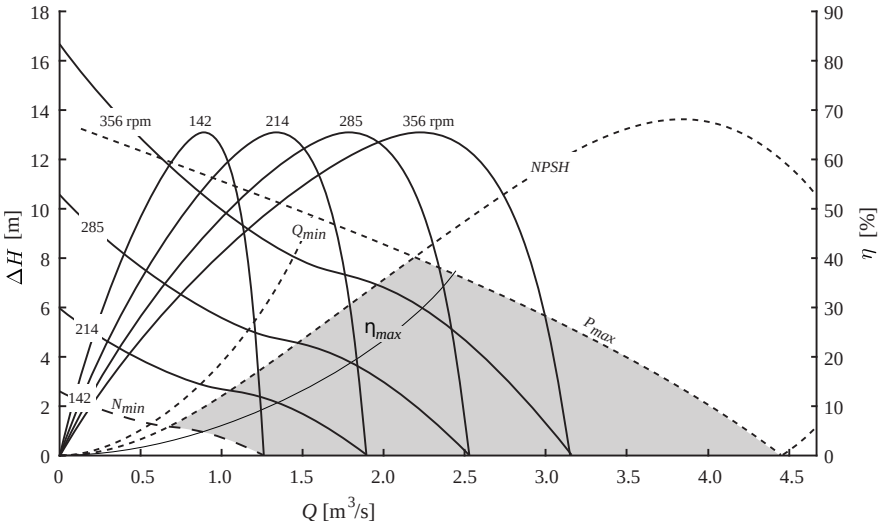


Figure 3.3 – Performance of the axial flow pump for different shaft speeds. The working area is bounded by minimum discharge Q_{min}, the minimum shaft speed N_{min}, cavitation NPSH, and maximum motor power P_{max}, and is depicted as a gray area.

3.2.2 Convex modeling and optimization of a pump that is always on

This section describes how the operation of the variable-speed pump can be modeled with convex functions in case the pump is switched on.

Consider an optimization problem

$$\begin{aligned} \min_x \ & f_0(x) \\ \text{s.t. } & f_i(x) \leq 0, \quad i = 1, \ldots, m, \\ & h_j(x) = 0, \quad j = 1, \ldots, p, \end{aligned} \tag{3.1}$$

where $x \in \mathbb{R}^n$ is the optimization variable, $f_0 : \mathbb{R}^n \mapsto \mathbb{R}$ is the objective function, $f_i : \mathbb{R}^n \mapsto \mathbb{R}$ are the inequality constraints and $h_j : \mathbb{R}^n \mapsto \mathbb{R}$ are the equality constraints. If the functions f_0 and f_i, $i = 1, \ldots, m$, are convex, and the functions h_j, $j = 1, \ldots, p$, are affine, then the optimization problem (3.1) is said to be convex (Boyd and Vandenberghe 2004). The next step is to approximate the operation of the pump in such form.

For ease of notation, we consider a single pump and a single spatial water level discretization point H_{up} only. Practical applications would typically involve a multitude of spatial discretization points along a channel system. The method presented in this chapter extends naturally to such cases.

The optimization problem for pumps is the following: how to choose the pump speed for the instantaneous head during a time period T so that the minimal amount of energy is used, while the upstream water level H_{up} is kept between its lower bound $\underline{H}_{up}$ and the upper bound $\overline{H}_{up}$. Using an equidistant discretization in

time, the optimization problem can be expressed as follows:

$$\min_{Q,H_{up},\Delta H,N} \sum_{k=1}^{T} P(N(t_k),\Delta H(t_k))\Delta t \tag{3.2}$$

$$\text{s.t. } \underline{H}_{up} \le H_{up}(t_k) \le \overline{H}_{up}, \qquad k=1,\dots,T, \tag{3.3}$$

$$\frac{wl}{\Delta t}(H_{up}(t_k) - H_{up}(t_{k-1})) \tag{3.4}$$

$$= Q_0(t_k) - Q(t_k), \qquad k=1,\dots,T, \tag{3.5}$$

$$\Delta H(t_k) = H_{down}(t_k) - H_{up}(t_k), \quad k=1,\dots,T, \tag{3.6}$$

$$g_i(Q(t_k),\Delta H(t_k),N(t_k)) \le 0, \qquad k=1,\dots,T,$$

$$i=1,\dots,m, \tag{3.7}$$

$$h(Q(t_k),\Delta H(t_k),N(t_k)) = 0, \qquad k=1,\dots,T, \tag{3.8}$$

where t_k is the discretized time, N is the pump speed, Q is the pump discharge, P is the power, ΔH is the static head, w is the width of the upstream reach, l is the length of the upstream reach, Δt is the time step of the discretization and T is the horizon of the optimization. The parameter $H_{up}(t_0)$ represents the fixed initial state of the system. The parameters $Q_0(t_k)$ and $H_{down}(t_k)$, $k=1,\dots,T$, are fixed time series describing the upstream inflow boundary condition and downstream level boundary condition, respectively. The linear constraint (3.5) describes the mass balance of the volume upstream of the pump. The linear constraint (3.6) relates the static head ΔH to the upstream water level H_{up} and the downstream level H_{down}. The inequality constraints (3.7) ensure that the pump operates inside its working area. The equality constraint (3.8) expresses the nonlinear $(Q,\Delta H)$-characteristic of the pump, which depends on the pump speed N following affinity theory.

In order for Problem (3.2) – (3.8) to be convex, the function P would have to be convex, the working area constraint functions g_i, $i=1,\dots,m$, would have to be convex, and the $(Q,\Delta H)$-

characteristics encoded in the equality constraint function h would have to be affine. However, none of these conditions are satisfied naturally.

Therefore, we will approximate Problem (3.2) – (3.8) with a new convex optimization problem. To do so, we take three steps: (1) discharge is taken as decision variable instead of pump speed, through which we can eliminate the equality constraint (3.8); (2) the inequality constraints (3.7) are approximated with convex inequality constraints $f_i(Q(t_k), \Delta H(t_k)) \leq 0$; and (3) a convex approximatation P_{app} is used instead of the original power consumption P. For a pump that is always switched on, we obtain

$$\min_{Q, H_{up}, \Delta H} \quad \sum_{k=1}^{T} P_{app}(Q(t_k), \Delta H(t_k)) \Delta t \tag{3.9}$$

$$\text{s.t. } \underline{H}_{up} \leq H_{up}(t_k) \leq \overline{H}_{up}, \qquad k = 1, \ldots, T, \tag{3.10}$$

$$\frac{wl}{\Delta t}(H_{up}(t_k) - H_{up}(t_{k-1})) \tag{3.11}$$

$$= Q_0(t_k) - Q(t_k), \qquad k = 1, \ldots, T, \tag{3.12}$$

$$\Delta H(t_k) = H_{down}(t_k) - H_{up}(t_k), \quad k = 1, \ldots, T, \tag{3.13}$$

$$f_i(Q(t_k), \Delta H(t_k)) \leq 0, \qquad k = 1, \ldots, T,$$

$$i = 1, \ldots, m. \tag{3.14}$$

Problem (3.9)–(3.14) has a convex quadratic objective, linear equality constraints, and convex quadratic inequality constraints.

In what follows the change of variables and the approximations are explained one by one.

The first step, the use of discharge instead of pump speed as decision variable, allows us to eliminate the pump speed variable N from the optimization problem, and therefore also allows us to eliminate the equality constraint (3.8). The computation of pump speeds is deferred to the post-processing stage, where it is

calculated using the unique relationship $N(Q, \Delta H)$ stated by the affinity rules:

$$\frac{Q_b}{Q_a} = \frac{N_b}{N_a}, \tag{3.15}$$

$$\frac{\Delta H_b}{\Delta H_a} = \left(\frac{N_b}{N_a}\right)^2, \tag{3.16}$$

where N_a and N_b are shaft speeds, and Q and ΔH are the corresponding discharge and head. A consequence of these affinity rules is that each $(Q, \Delta H)$ point in the working area is related to a pump speed via a parabolic relation. An example for such a calculation is given in Appendix 3.A.

The second step is to approximate the non-convex inequality constraints (3.7) with convex inequality constraints (3.14). This is implemented in the following way: the bounding curves of the working area are approximated with quadratic functions. If such an approximation is not convex, then it is replaced with a linear approximation. These functions are then used as inequality constraint functions in (3.14). An example is shown in Figure 3.4. The dashed lines approximate the boundaries of the working area: the maximum power curve, the NPSH curve and the minimum shaft speed curve. The maximum power curve is approximated with a quadratic function, which is convex in Q. The NPSH and the minimum shaft speed curve should be concave in Q, and thus they are approximated with a linear function. The convex working area (dashed lines) is only slightly different from the original one (gray area). This means that the optimizer can select somewhat different choices of $(Q, \Delta H)$ points at some of the boundaries, than it would when using a non-convex working area.

The third change is the convex approximation of the power. The power is a function of two variables: the discharge and the head.

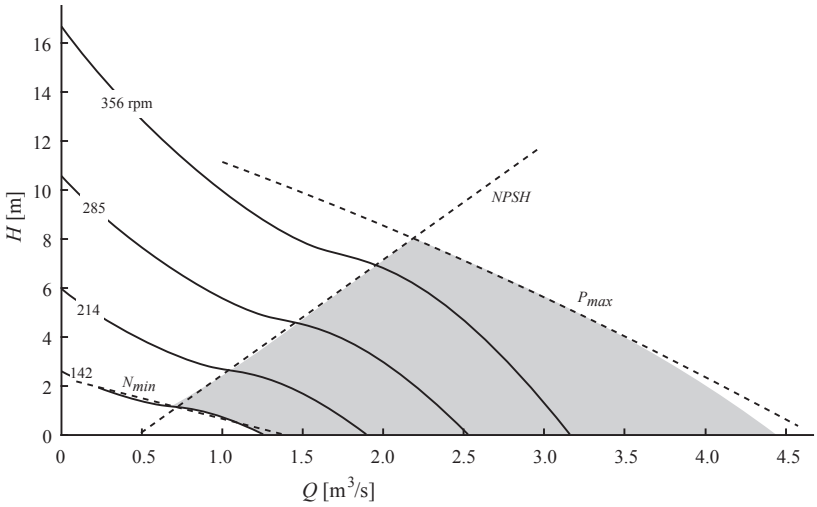


Figure 3.4 – The working area of the pump. The original area is shaded with gray, and the convex quadratic approximation of the bounds is shown with dashed lines.

It can be approximated by a quadratic function

$$P_{app} = P_{c00} + P_{c10}Q + P_{c01}\Delta H + P_{c20}Q^2 + 2P_{c11}Q\Delta H + P_{c02}\Delta H^2.$$

This function is convex if its Hessian with respect to $(Q, \Delta H)$ is positive semi-definite.

The approximation of the power is calculated as follows. First the power P_i is determined in different points $(Q_i, \Delta H_i)$ in the working area on the basis of data sheets and/or experiments. Then the coefficients P_{cij} are obtained through the following optimization

problem:

$$\min_{P_{cij}} \sum_i (P_{c00} + P_{c10}Q_i + P_{c01}\Delta H_i \tag{3.17}$$

$$+ P_{c20}Q_i^2 + 2P_{c11}Q_i\Delta H_i + P_{c02}\Delta H_i^2 - P_i)^2 \tag{3.18}$$

$$\text{s.t.} \begin{pmatrix} P_{c20} & P_{c11} \\ P_{c11} & P_{c02} \end{pmatrix} \succeq 0. \tag{3.19}$$

The convexity of the power approximation is guaranteed by constraint (3.19), which states that the Hessian must be positive semi-definite. The set of positive semi-definite matrices of a given size forms a (convex) cone (Boyd and Vandenberghe 2004, e.g.), whence the optimization problem (3.18)–(3.19) itself is also convex.

In the following, the error of the resulting power approximation is assessed. The error between the power and the approximated power did not exceed 10% for the case considered in this study. An error of 10% is, however, relatively large. This is a factor that has motivated the development of the invexity approach in Chapter 5.

3.2.3 Mixed-integer optimization: including the possibility to switch off the pump

In this section, the optimization problem is extended to the case when the pump may be switched off. This can be described as a mixed-integer optimization problem, where at least one of the variables can only take integer values. An approximate solution to such a problem may be found by solving a relaxation to a continuous problem, in which the integer variables are allowed to vary continuously. In order for a typical solver to be able to find a global optimum of the mixed-integer problem, these relaxed problems should be convex. Such solvers typically solve

many relaxations over the course of a solution procedure (*e.g.*, in a *branch-and-cut* scheme (Floudas 1995)), which can lead to high computation times. The convexity of the continuous relaxations of the mixed-integer problem will be guaranteed by making sure that the integer variables enter into the constraints linearly using a so-called *big-M* reformulation (Floudas 1995, e.g), as explained below.

In order to describe the pump optimization as a mixed-integer problem we introduce $\delta(t_k)$, a boolean variable indicating if the pump is on ($\delta(t_k) = 1$) or off ($\delta(t_k) = 0$) at time t_k. Suppose the power is approximated with a convex polynomial $P_{app}(Q, \Delta H)$ which is not necessarily zero at zero discharge. Then, it can occur that when the discharge is zero, the consumed power is nevertheless estimated to be positive. If the pump is allowed to be switched off, P_{app} should therefore be multiplied by $\delta(t_k)$, so that it is zero when the pump is off. However, the product of these two variables does not necessarily lead to convex relaxations. This optimization problem can be written as:

$$\min_{Q, H_{up}, \Delta H, \delta} \sum_{k=1}^{T} \delta(t_k) P_{app}(Q(t_k), \Delta H(t_k)) \Delta t \tag{3.20}$$

$$\text{s.t. } \underline{H}_{up} \leq H_{up}(t_k) \leq \overline{H}_{up}, \qquad k = 1, \ldots, T, \tag{3.21}$$

$$\frac{wl}{\Delta t}(H_{up}(t_k) - H_{up}(t_{k-1})) \tag{3.22}$$

$$= Q_0(t_k) - Q(t_k), \qquad k = 1, \ldots, T, \tag{3.23}$$

$$\Delta H(t_k) = H_{down}(t_k) - H_{up}(t_k), \quad k = 1, \ldots, T, \tag{3.24}$$

$$f_i(Q(t_k), \Delta H(t_k)) \delta(t_k) \leq 0, \qquad k = 1, \ldots, T,$$

$$i = 1, \ldots, m, \tag{3.25}$$

$$\delta(t_k) = \begin{cases} 1 & \text{if } Q(t_k) > 0, \\ 0 & \text{otherwise,} \end{cases} \qquad k = 1, \ldots, T. \tag{3.26}$$

In order to transform Problem (3.20)–(3.26) to a problem whose

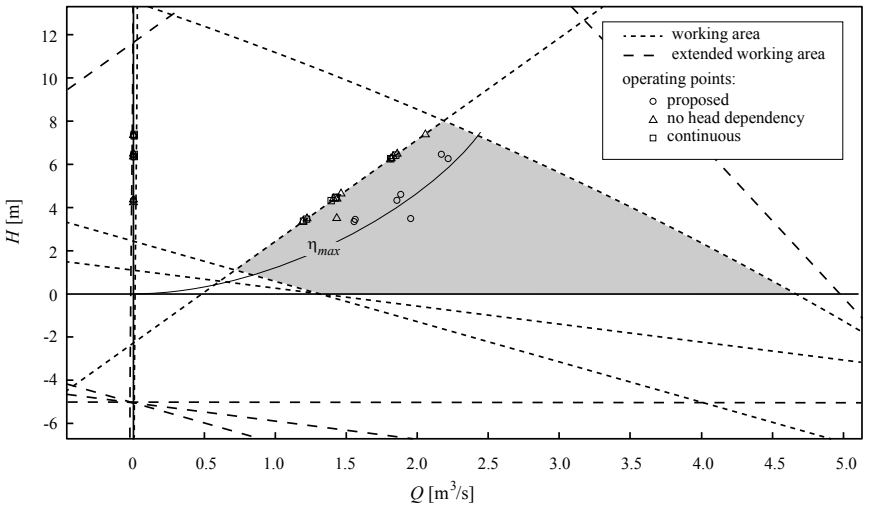


Figure 3.5 – Definition of the extended working area (dashed lines). The extended working area is shown in case the pump is switched off. The markers show the actual operating points of the pump during the presented case study.

continuous relaxation is convex, the objective function and the constraints need to be replaced with convex functions.

First, the transformation of the working area constraints (3.25) is described, and subsequently the transformation of the objective function (3.20). The working area constraints define the working area of the pump when it is on: the working area is shaded and its boundaries are shown with dashed lines in Figure 3.5. If the pump is off, the discharge is zero, but the head is not, therefore the $(Q, \Delta H)$ points will be on the ΔH-axis. However, this is outside the working area. In order to allow the pump to be off, the working area should be extended to contain the physically feasible portion of the ΔH-axis when the pump is off. This is achieved by shifting the left boundaries of the working area. The constraints should be

shifted such that they include the ΔH-axis with the minimum and maximum possible head, ΔH_{min} and ΔH_{max}, respectively. These head values are obtained from the minimum and maximum water levels of the physical system.

Next, an offset value $f_{\text{offset,i}}$ is computed that satisfies the relation

$$\begin{aligned} f_{\text{offset,i}} \geq \max_{Q,\Delta H} & f_i(Q, \Delta H) \\ \text{s.t.} & (Q, \Delta H) \in E, \end{aligned} \tag{3.27}$$

where E denotes the extended working area. This maximization problem is, in general, non-convex, whence an offset $f_{\text{offset,i}}$ is typically chosen to overestimate the maximum. Alternative approaches include the use of adjustable robust optimization to compute a tight upper bound (Selvi et al. 2020).

Once an appropriate offset value has been determined, the working area constraints (3.25) may be rewritten as:

$$f_i(Q(t_k), \Delta H(t_k)) \leq (1 - \delta(t_k)) f_{\text{offset,i}}, \quad i = 1, \ldots, m. \tag{3.28}$$

This inequality constraint is convex. If the pump is on ($\delta(t_k) = 1$), it is equivalent to the original convex constraint (Equation (3.14)). If the pump is off ($\delta(t_k) = 0$), the constraint is satisfied thanks to the choice of $f_{\text{offset,i}}$ as per Equation (3.27). This formulation makes it possible to extend the working area to the situation when the discharge is zero and the pump is off.

Now the transformation of the objective function from non-convex to convex is described. Currently it is a convex function multiplied with a boolean, an expression which is not necessarily convex. For this, we will use a big-M reformulation (Floudas 1995, e.g). As a first step, we move the approximated power to the constraints by replacing P_{app} in the objective with an auxiliary variable P_{help}. The sum of this single variable over time is going to be minimized

and a new constraint is added:

$$P_{app}(Q(t_k), \Delta H(t_k)) \leq P_{help}(t_k). \tag{3.29}$$

As P_{help} is minimized, its possible smallest value will be equal to P_{app}. The objective function in Problem (3.20)–(3.26) is multiplied with a boolean. This step was necessary to express that the power is zero when the pump is off, while the power approximation function is not zero in that case. In order to create a convex constraint, the function P_{app} should not be multiplied with a boolean, but we still need to get $P_{help}(t_k) = 0$ whenever $Q(t_k) = 0$. The following formulation achieves this. Let us denote the minimum and maximum of the approximated power by m and M respectively. These values can be obtained from the physical characteristics of the pump. The power constraint (3.29) is replaced by the following three inequalities:

$$\begin{aligned}
&0 \leq Q(t_k) \leq \delta(t_k)\overline{Q}, \\
&m\delta(t_k) \leq P_{help}(t_k) \leq M\delta(t_k), \\
&P_{app}(Q(t_k), \Delta H(t_k)) \leq P_{help}(t_k) + M(1 - \delta(t_k)).
\end{aligned}$$

According to these constraints, if $\delta(t_k) = 0$, the pump is off, $0 \leq P_{help}(t_k) \leq 0$, so $P_{help}(t_k) = 0$. If $\delta(t_k) = 1$, the pump is on, $m \leq P_{help}(t_k) \leq M$ and $P_{app}(t_k) \leq P_{help}(t_k)$ is equivalent to the original constraint.

The final optimization problem for a variable-speed pump, valid

for all conditions, is:

$$
\begin{aligned}
\min_{Q,H_{up},\Delta H,\delta} \ & \sum_{k=1}^{T} P_{help}(t_k)\Delta t \\
\text{s.t. } & \underline{H}_{up} \leq H_{up}(t_k) \leq \overline{H}_{up}, && k = 1,\dots,T, \\
& \frac{wl}{\Delta t}(H_{up}(t_k) - H_{up}(t_{k-1})) \\
& \quad = Q_0(t_k) - Q(t_k), && k = 1,\dots,T, \\
& \Delta H(t_k) = H_{down}(t_k) - H_{up}(t_k), && k = 1,\dots,T, \\
& f_i(Q(t_k), \Delta H(t_k)) \\
& \quad \leq (1 - \delta(t_k)) f_{\text{offset},i}, && k = 1,\dots,T, \\
& && i = 1,\dots,m, \\
& P_{app}(Q(t_k), \Delta H(t_k)) \\
& \quad \leq P_{help}(t_k) + M(1 - \delta(t_k)), && k = 1,\dots,T, \\
& m\delta(t_k) \leq P_{help}(t_k) \leq M\delta(t_k), && k = 1,\dots,T, \\
& 0 \leq Q(t_k) \leq \delta(t_k)\overline{Q}, && k = 1,\dots,T, \\
& \delta(t_k) \in \{0,1\}, && k = 1,\dots,T.
\end{aligned}
\tag{3.30}
$$

Problem (3.30) is a mixed-integer optimization problem with convex quadratic objective and constraints. The number of optimization variables depends on the optimization horizon and the temporal discretization step size Δt.

For the mixed-integer formulation, several auxiliary parameters are used (such as m, M, $\overline{Q}$). These parameters must satisfy the inequalities described above. If these parameters are, however, estimated too conservatively, *e.g.*, by choosing a very small m or a very large M, then the computation time of the solution is negatively affected. This is due to the fact that in this case, the solutions to the continuous relaxations will have objective function values that are much lower than those that can be obtained

using mixed-integer decisions. Such "loose" relaxations reduce the effectivity of branch pruning (Floudas 1995, e.g.) in mixed-integer solvers.

3.2.4 The case of constant-speed pumps

The methodology described above can be applied to constant-speed pumps. The working area of a constant-speed pump is a special case: the curves defining the minimum and maximum shaft-speed coincide, and the area is reduced to a line. An example of the working area of a constant-speed pump is shown in Figure 3.6. The curves bounding the working area define the beginning and end point of the segment, with other words the minimum and maximum possible discharge. The pump can operate either on this segment, or, in case it is off, the vertical axes. The markers show examples of operation of a constant-speed pump. As the minimum and maximum shaft speed lines coincide, the equation serves as minimum and maximum bound, thus this constraint has to be affine for the relaxations to be convex, which is in fact a new level of approximation.

3.3 Results and discussion

The proposed method can be implemented in an MPC formulation with a receding horizon. After each time step, the horizon shifts and the optimization is performed again. In this section we demonstrate one single optimization as an indication of performance. The numerical example is shown and compared with two other methods in order to demonstrate the advantages of our approach. First, we introduce the water system and the case study.

This water system consists of a lower lying area (polder) and an area with higher water level (sea). The goal is to keep the water

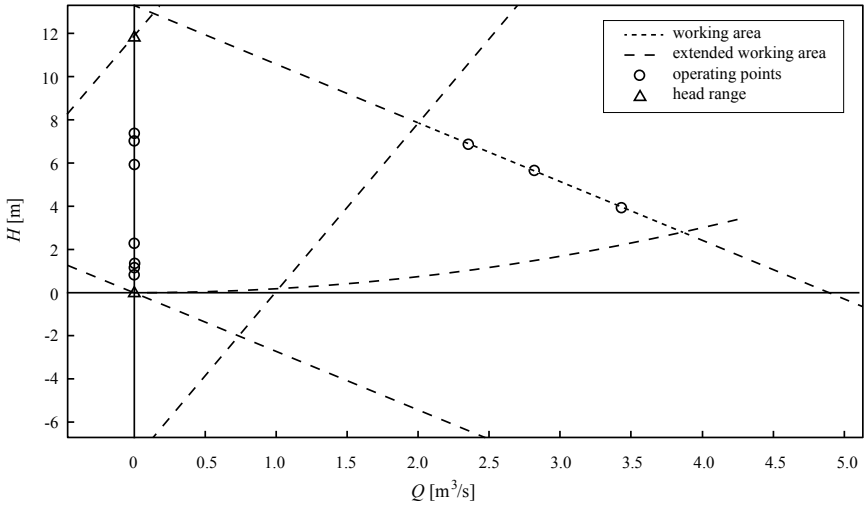


Figure 3.6 – Definition of the extended working area of a constant-speed pump. The extended working area is shown in case the pump is switched off. The circles show the actual working point of the pump during the presented case study.

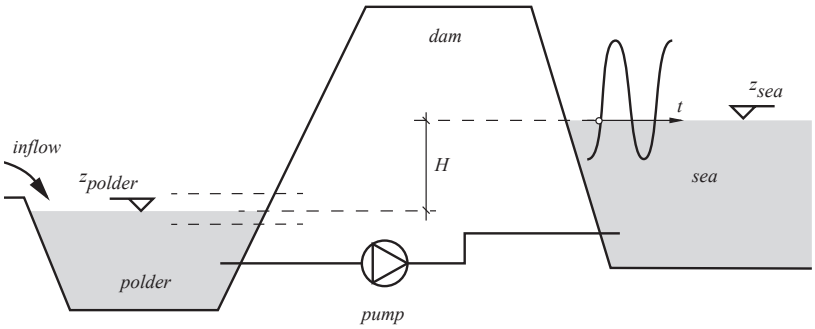


Figure 3.7 – Schematic of the case study.

level of the polder within some pre-defined bounds by pumping while water is flowing into the polder. A schematic view of the system is shown in Figure 3.7. The polder water level is at the midpoint between the lower and upper bounds at the beginning of the period. Water is flowing into the polder, while the sea water level also changes due to the tidal motion. A terminal constraint forces the water level to reach the midpoint between the lower and upper bounds at the end of the time-period. In order to reach the goal the excess water should be pumped out by consuming as little energy as possible.

Without pumping, the water level would rise to exceed the maximum polder level. It is favorable to wait for the polder water level to rise and/or the sea water level to drop (low tide) before starting pumping, as a lower head will lead to a lower energy consumption. However, letting the water level increase excessively in the polder might also risk exceeding the maximum level. In what

follows, the pump operation is calculated by optimization using three different methods: the method proposed in this chapter, a method using head-independent power (Menke et al. 2016), and a method using continuous optimization (Ulanicki and Kennedy 1994). The time horizon is 34 hours in all cases. The time step is 2 hours.

Our mixed-integer formulation is solved using Bonmin (Bonami et al. 2008) wrapped in RTC-Tools 2, an open-source toolbox for control and optimization of water systems (Baayen et al. 2018).

The results calculated by the proposed method are shown in Figure 3.8. It can be seen that the objective is met: the polder water level stays within the prescribed bounds, while the energy used by the pump is 1120 kWh (Table 3.1). The pump switches on at the start of the period, and switches off after two hours when the tide and therefore the head across the pump is highest. Then it switches on again and pumps until the water level almost reaches the minimum level, thus taking advantage of the entire low tide period. The final water level reaches the midpoint between the lower and upper bounds as a result of the terminal constraint. The operating points of the pump are shown in Figure 3.5 with diamonds. It can be seen that the pump operates close to the maximum efficiency whenever possible.

A second set of computations was carried out with a mixed-integer optimization method similar to the proposed one, but in this case the power was modelled as quadratic function of the discharge alone, *i.e.*, without depending on the head ΔH. Figure 3.9 shows that in this case the pump switches on in the beginning of the period and it keeps on pumping even during the highest tide. It stops when the water level is close to the minimum level, and it restarts when the polder is full. As in this formulation the power does not depend on the head, the pump prefers to operate

Table 3.1 – Energy consumption and computation time of the different methods.

	Proposed method	**No head dependency**	**Continuous**
Energy (kWh)	1120	1274	1235
Time (s)	660	230	60

at low discharges rather than at low head. This leads to less efficient pumping: it can be seen that the duty points of this pump (triangles in Figure 3.5) are further from the best efficiency line and predominantly at minimum discharge. This approach was also to able to meet the objective: the polder water level stayed within the prescribed bounds. However, the energy consumption was 1274 kWh, which is about 14% higher than that of the proposed method (Table 3.1).

Several practitioners suggest to use continuous optimization, and to then round the results to booleans that indicate when the pump is on or off. The advantage of this approach is the reduced computation time. Therefore, we also carried out a third set of computations with a continuous algorithm. The rounding of the solution took place step by step: the optimization was ran once, then the first decision was rounded and fixed, then it was ran again for the remainder of the horizon, etc. In this case the pump does not switch on in the beginning of the period, but only after 4 hours (Figure 3.10). Then it pumps until the second high tide arrives. The water level remains within bounds throughout the optimization horizon, and in the end it reaches the midpoint of the operating level range. The pump consumed 1235 kWh, *i.e.*, about 10% percent more than the proposed method.

Comparing the energy consumption of the three methods (Table 3.1), it is clear that the proposed method outperforms the existing methods for this case study: the energy consumption is

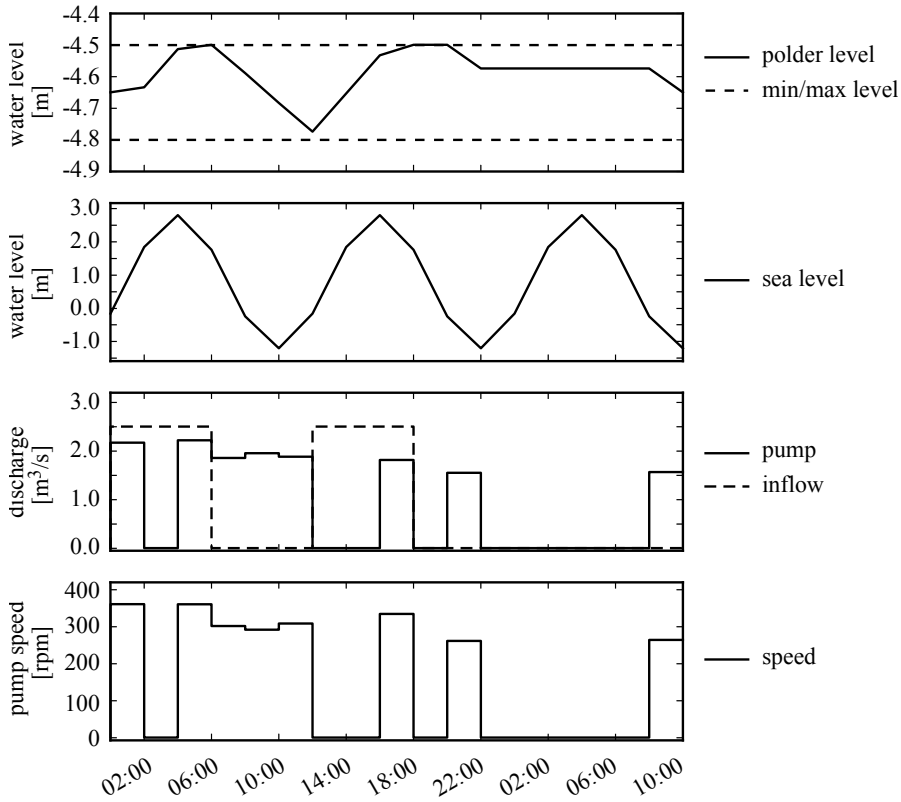


Figure 3.8 – Results with the proposed optimization system.

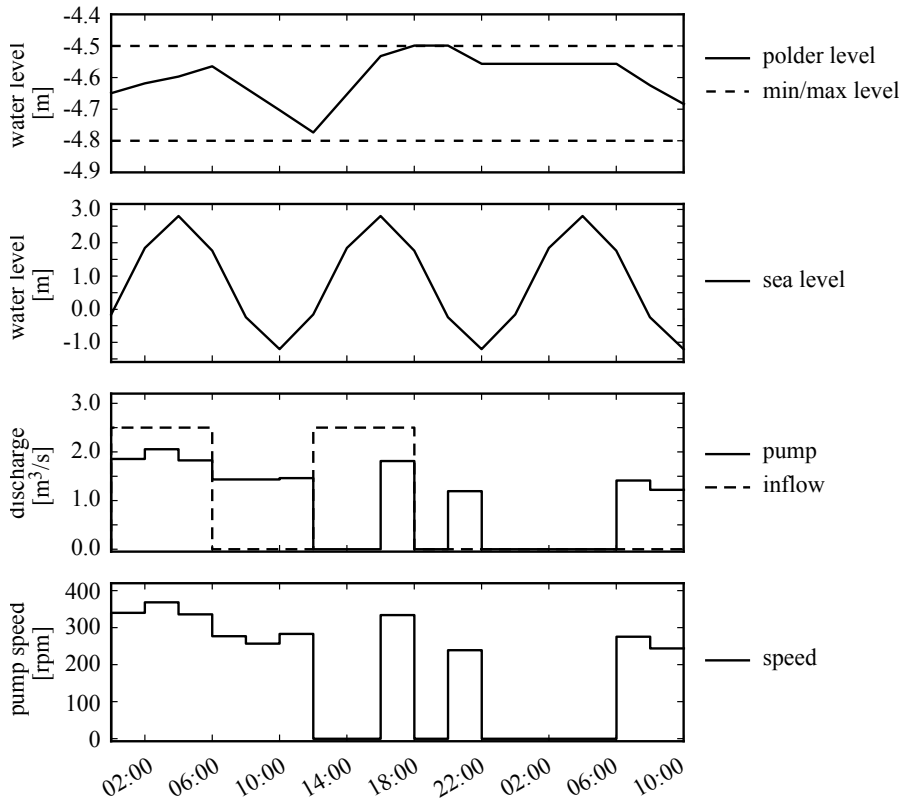


Figure 3.9 – Results without head dependency.

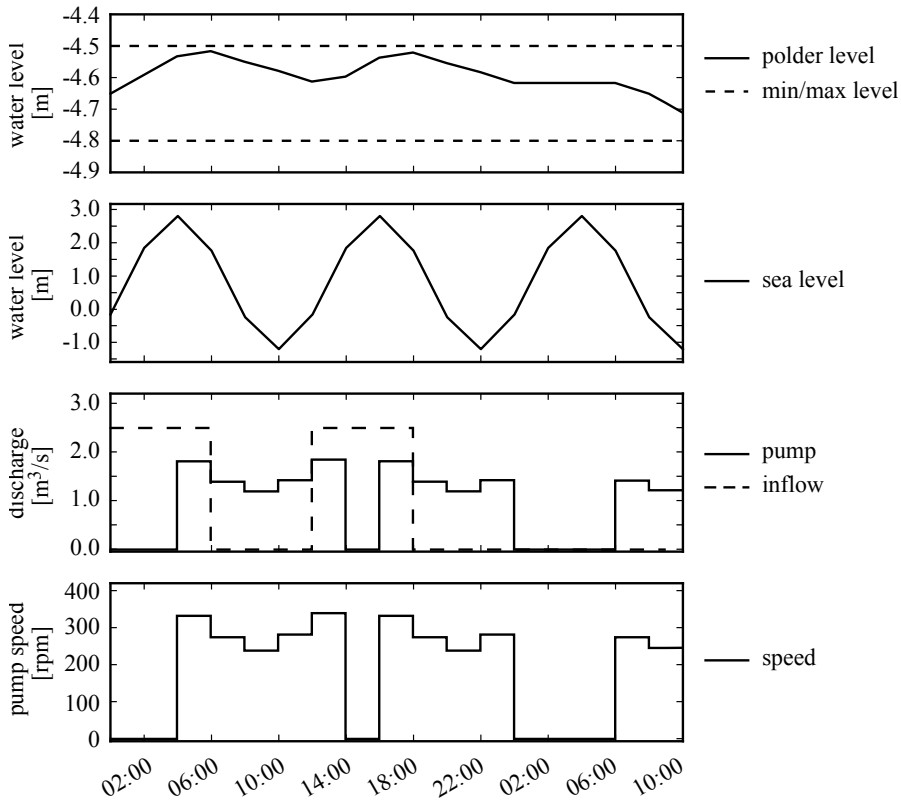


Figure 3.10 – Results with continuous optimization.

lower compared to the other two methods. All methods meet the desired objective of keeping the water level within the given level operating range.

It should be noted that the calculation time of the continuous method is almost an order of magnitude lower than for the other two mixed-integer methods. For this example, however, the run times of all three methods are acceptable for real-time implementation. It should be considered that by increasing the size of the problem, *i.e.* when including more pumps, the proposed method might lead to prohibitive computation times. As this example demonstrates, for problems consisting of small numbers of pumps, our approach is promising.

The method can be used for different kinds of pumps and for different applications. It can be applied for water distribution networks with storage capacity, including water tanks or water towers. In that case the water levels would be replaced by pressure heads in the problem formulation. It can be applied for pumped-storage where the driving parameter is energy price. The objective function can also be extended by including a constant energy price time series.

3.4 Conclusions

In this chapter a convex approximation to the pump scheduling problem was introduced. The convex approximation method has several advantages compared to existing heuristic methods: it finds a global optimum, it includes pump switches, and it is applicable to variable-speed pumping. Due to these advantages, more energy can be saved by using this method when performing optimization. The methodology and its advantages were demonstrated on a numerical example for a drainage pump. In this

example, the energy use is shown to be 10 to 14 percent lower in comparison with results from other optimization methods. A further implication of the method is that not only energy use can be optimized, but also energy costs or use of renewable energy. The method can be used in several applications of pumps such as polder systems, ground water pumping systems, pumped-storage systems and water transmission or distribution networks with water towers or storage tanks.

3.4 Acknowledgments

This research was carried out within the *Slim Malen* (Smart Drainage) project. The Slim Malen project was funded and carried out by the following partners: STOWA, Ministry of Economic Affairs (RVO), Deltares, Eindhoven University of Technology, Ministry of Infra and Environment (Rijkswaterstaat – WVL), Water Authorities Hoogheemraadschap Hollands Noorderkwartier, Fryslân, Zuiderzeeland, Rivierenland, Scheldestromen, Rijnland, Brabantse Delta and Hollandse Delta and by companies Nelen & Schuurmans, e-Risk Group, Eneco, Delta, Alliander EXE, Actility, XYLEM and KISTERS. The calculations are carried out by the open source software RTC-Tools 2 (Baayen et al. 2018). The data for the case study can be found at `https://zenodo.org/record/1880457#.XAUGMzhKiUk`.

Appendix

3.A Calculation of pump speed using affinity rules

In this appendix we describe an example calculation of the pump speed in order to run at a specific working point $(Q, \Delta H)$, using affinity rules. The pump has a $(Q, \Delta H)$-curve as given in Table 3.2 for a reference speed of 356 rpm.

Table 3.2 – Pump performance curve at reference speed of 356 rpm.

Q	ΔH
0.041	16.322
0.273	14.42
0.546	12.577
0.897	10.544
1.234	8.769
1.597	7.592
1.943	6.911
2.253	5.848
2.517	4.612
2.768	3.014
3.034	1.062

This performance curve is shown in Figure 3.11. It is the same pump as presented in the chapter, however with a lower number of points for simplicity. Suppose the result of the optimization is that the pump should operate at the working point (Q_{wp},ΔH_{wp})=(2.39 m^3/s, 4 m). The pump speed can be found via post-processing using the affinity rules as follows. First, the working point is scaled by using the following relation resulting from the combination of

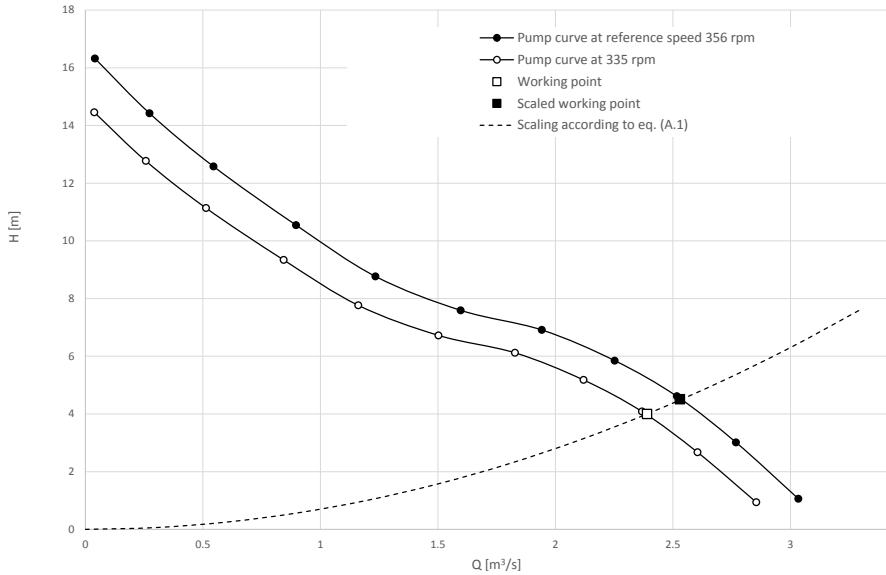


Figure 3.11 – Scaling of reference pump curve for operation at the working point.

Equations (3.15) and (3.16):

$$\frac{\Delta H_b}{\Delta H_a} = \left(\frac{Q_b}{Q_a}\right)^2 .$$

For $H_a = H_{wp}$ and $Q_a = Q_{wp}$, this relation is shown in Figure 3.11 as the dashed curve. From the graph it shows that the corresponding working point at reference speed is at a head of 4.51 m and a discharge of 2.53 m^3/s. Now using Equation (3.16) with ΔH_a = 4.51 m, ΔH_b = 4 m, N_a = 356 rpm, the new shaft speed N_b is calculated as 335 rpm. The scaled pump performance curve at 335 rpm can be calculated using Equations (3.15)-(3.16) and is presented in Table 3.3 and shown in Figure 3.11.

Table 3.3 – Scaled pump performance curve at 335 rpm.

Q	ΔH
0.038	14.453
0.257	12.769
0.513	11.137
0.844	9.337
1.161	7.765
1.503	6.723
1.828	6.12
2.12	5.179
2.368	4.084
2.605	2.669
2.855	0.94

CHAPTER 4

Closed-loop model predictive control with mixed-integer optimization of a river reach with weirs

4.1 Introduction

Optimization is a powerful tool to assist the management of water systems. Model predictive control (MPC) is especially suitable for this purpose due to the following properties: (1) it is able to proactivately act on forecasted disturbances, (2) it is able to handle time delay, and (3) the control actions are re-calculated and adjusted at every control time step to include newly available data (*e.g.*, a new inflow forecast or level measurement). Many studies have been carried out on applications of MPC to water systems (Rodellar et al. 1989, Wahlin 2004, van Overloop 2006b, e.g.). In these studies linear models are used to preserve convexity even though the problem at hand is essentially non-convex. By relaxing nonlinear equality constraints to inequality constraints, however, it is sometimes possible to formulate a convex optimization problem that preserves the nonlinearity of the system model.

When water systems are controlled by under- or overshot gates (*i.e.*, weirs, see Figure 4.1), their dynamics should be included into

the system. When doing so, a control variable should be selected. For decentralized control schemes, control of the gate position has the advantage that it is possible to take the dynamics of the gate actuator into account (Malaterre and Baume 1999).

Every gate setting and water level combination generates a particular discharge. If one chooses the resulting discharge as optimization variable and disregards the water level dynamics, one obtains the advantage that the river reaches become decoupled. It is, however, more accurate to use the gate position as control variable (Horváth et al. 2013). Weyer (2006) also uses discharge as control variable to keep the model linear.

This chapter focuses on linear MPC. In order to have a linear model either the weir equation should be linearized (Horváth et al. 2013), or the crest level of the weir should be calculated via post-processing. In the second case the control variable is the discharge. In the first case, the physical limitations of the system (*i.e.*, minimum and maximum crest level) can be included directly as constraints. In the second case, however, the crest level is no longer a control variable, and the translation of the physical constraints to the discharge variable leads to a hybrid system that can be optimized using mixed-integer optimization.

We present a mixed-integer weir modeling approach using discharge as control variable, including an approximation of the physical constraints of the crest level. Our approach is such that the continuous relaxations of the mixed-integer problems are convex. The continuous part of this approach is similar to the pump modelling of Chapter 3. In the present chapter the approach is extended to a mixed-integer setting, where no flow over the weir is allowed. Additionally, a closed-loop testing is presented, in which the decisions from the optimization are sent to a hydrodynamic model of the water system, and the results of that model are sent

Figure 4.1 – Weir on the Vecht river, the Netherlands. Source: Wikipedia.

back to the optimization as the new initial state. The approach is implemented in RTC-Tools 2, the open-source water systems optimization package developed at Deltares (Baayen et al. 2018). The goal of this chapter is to present a mixed-integer approach for weir-flow optimization and to demonstrate its advantages by comparing it with the continuous approach.

4.2 Material and methods

4.2.1 Convex optimization approach

The RTC-Tools 2 optimization system offers a convex optimization option, which guarantees that for appropriately formulated problems, a global optimum is reached. This property is crucial for a decision support system. If the problem were not convex, a local optimum might be reached instead, and a small change in the initial conditions might direct the solution into an entirely different local optimum. This fact would reduce the credibility of the decision support system to the user. Therefore, we aim at approximating the water problem as a convex optimization problem

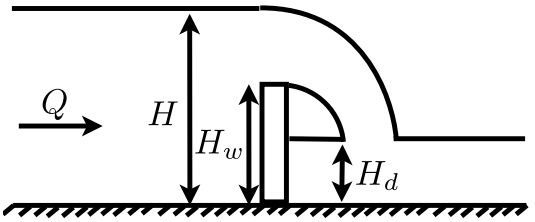


Figure 4.2 – Schematic view of a weir, where Q is the upstream discharge, H is the upstream water level, H_w is the crest level of the weir, and H_d is the downstream water level.

in the form (Boyd and Vandenberghe 2004):

$$\begin{aligned} &\min_x \ f_0(x) \\ &\text{s.t. } f_i(x) \leq 0 \quad i = 1, \ldots, m, \\ &\qquad a_j^T x = b_j \quad j = 1, \ldots, p, \end{aligned}$$

where $f_0, \ldots, f_m$ are convex functions: f_0 is the objective function and $f_1, \ldots, f_m$ are the inequality constraint functions. $x \in \mathbb{R}^n$ is the optimization variable, m and p are the number of inequality and equality constraints, respectively.

4.2.2 Modeling of the water flow

The water movement in the reaches is modelled with an integrator-delay model:

$$wl\frac{dH}{dt}(t) = Q_{in}(t) - Q_{out}(t + \tau),$$

where Q_{in} and Q_{out} are the in- and outflow to the reach, H is the water level, τ is the time delay, w is the width of the river reach, and l its length.

4.2.3 Weir modeling

Weir discharge is computed from the physical characteristics of the weir and from the water depth upstream (Figure 4.2). If the upstream water level H is higher than the crest H_w and also higher than the downstream water level H_d, there is flow over the weir in downstream direction. In this direction there are two operation modes: if the upstream water level H is higher than the downstream water level H_d, water flows freely over the weir crest H_w. This is called the free flow regime. If the downstream water level H_d is high enough, the weir is said to be submerged. In this chapter only the free flow regime is treated, in a single flow direction. This flow regime is described using the common weir equation (Te Chow 1959):

$$Q = \begin{cases} C_w(H - H_w)^{3/2} & \text{if } H > H_w, \\ 0 & \text{if } H \leq H_w, \end{cases} \tag{4.1}$$

where

$$C_w = C_d \frac{2}{3} \sqrt{2g} B_w,$$

and where C_w contains all the constants in the weir equation, C_d is the weir discharge coefficient (typically around 0.61, based on Sepúlveda (2008)), g is the acceleration of gravity, B_w is the width of the weir, H_w is the crest level, and H is the upstream water level. The equation is plotted in Figure 4.3 in black: the left curve is the weir equation evaluated with the minimum crest level, the right curve is evaluated with the maximum crest level. Figure 4.3 shows that until a certain water level, the discharge remains zero. This occurs when the water level is not reaching the crest level. When the water level reaches the crest level, the discharge becomes proportional to the 1.5^{th} power of the the difference between the water level and the crest level.

4.2.4 The development of the optimization problem

In this section the mixed-integer optimization problem is described. First, the continuous non-convex optimization problem is shown. Next, the steps are explained to arrive at a mixed-integer convex optimization problem.

Using the crest level as optimization variable, it is straightforward to impose constraints on the crest level of the weir. The resulting optimization problem, however, is not convex. The discretized optimization problem for an open water channel of m reaches divided by weirs can be written as:

$$
\begin{aligned}
&\min_{H_w,Q,H} \ 1 \\
&\quad \text{s.t. } \underline{H}_i \le H_i(t_k) \le \overline{H}_i, && k = 1,\dots,T, \\
&&& i = 1,\dots,m, \\
&\qquad \frac{w_i l_i}{\Delta t}(H_i(t_k) - H_i(t_{k-1})) \\
&\qquad\quad = Q_{i-1}(t_k) - Q_i(t_k), && k = 1,\dots,T, \\
&&& i = 1,\dots,m, \\
&\qquad \underline{H}_{w,i} \le H_{w,i}(t_k) \le \overline{H}_{w,i}, && k = 1,\dots,T, \\
&&& i = 1,\dots,m, \\
&\qquad Q_i(t_k) = C_{w,i}(H_i(t_k) - H_{w,i}(t_k))^{3/2}, && k = 1,\dots,T, \\
&&& i = 1,\dots,m,
\end{aligned}
\tag{4.2}
$$

where t_k is the discretized time, i is the spatial discretization point, Δt is the (uniform) length of the time step, and T is the prediction horizon. The parameters $H_i(t_0)$ represent the fixed initial state of the system, and $Q_0(t_k)$, $k = 1,\dots,T$, is a time series representing the upstream flow boundary condition. For ease of notation, the delay in the integrator-delay model has been set to $\tau = 0$. This is not restrictive as the integrator-delay model remains linear for

any value of τ. The objective is to find weir positions that keep the water level between the lower bound $\underline{H}$ and the upper bound $\overline{H}$. This is a feasibility problem. It can be extended to a minimization problem, for instance by including an objective function that penalizes a deviation from the target level.

In order to approximate Problem (4.2) as a convex optimization problem, the last constraint should be replaced. We do this by taking the discharge as the control variable, and computing the crest level via post-processing. The crest level variable is eliminated by splitting the weir equation into inequality constraints. The resulting intermediate non-convex optimization problem is the following:

$$\min_{Q,H} \ 1 \tag{4.3}$$

$$\text{s.t. } \underline{H}_i \leq H_i(t_k) \leq \overline{H}_i, \qquad k = 1, \ldots, T, \quad i = 1, \ldots, m, \tag{4.4}$$

$$\frac{w_i l_i}{\Delta t}(H_i(t_k) - H_i(t_{k-1})) \tag{4.5}$$

$$= Q_{i-1}(t_k) - Q_i(t_k), \qquad k = 1, \ldots, T, \quad i = 1, \ldots, m, \tag{4.6}$$

$$0 \leq Q_i(t_k) \leq \overline{Q}_i, \qquad k = 1, \ldots, T, \quad i = 1, \ldots, m, \tag{4.7}$$

$$C_{w,i}(H_i(t_k) - \overline{H}_{w,i})^{3/2} \leq Q_i(t_k), \qquad k = 1, \ldots, T, \quad i = 1, \ldots, m, \tag{4.8}$$

$$Q_i(t_k) \leq C_{w,i}(H_i(t_k) - \underline{H}_{w,i})^{3/2}, \qquad k = 1, \ldots, T, \quad i = 1, \ldots, m. \tag{4.9}$$

The last three constraints concern the weir. Constraint (4.7) bounds the discharge with a certain constant $\overline{Q}_i$. Note that this physically might not be necessary, but it will be needed for the model formulation, and in practice it is straightforward to give an

upper bound for the possible discharge. Constraint (4.8) implements a physical limitation: it limits the possible discharge with the maximum crest level, $\overline{H}_{w,i}$. The discharge cannot be less for a certain water level as it is permitted by the maximum crest level, (*e.g.*, there is too much flow and the weir starts to raise to close the flow, but the crest cannot be raised higher than physically feasible). Constraint (4.9) is related to the physical limitation of the minimum crest level, $\underline{H}_{w,i}$: the discharge corresponding to a certain water level cannot be higher than the minimum crest level permits. These three constraints are shown in Figure 4.3, and the gray area enclosed by these lines (both light and dark gray) is known as the working area. For any (Q_i, H_i) point from the working area, a crest level $H_{w,i}$ exists that realizes the flow Q_i.

This working area is, however, not convex, since (4.9) in not convex. Therefore, the left curve is approximated linearly. In order to enable the use of MILP solvers, also the right curve is linearized. The derivation of the two bounding approximating lines ($\underline{Q}_{w,i}$ and $\overline{Q}_{w,i}$) is detailed in Appendix 4.A. The approximation results in a convex working area, as illustrated in Figure 4.3.

Note that the approximation is conservative: the approximated working area lies completely within the nonlinear feasible area. This means that any resulting flow-head pair from within the working area has a corresponding crest level that is physically realizable and respects the nonlinear weir equation (4.1). On the other hand, the exclusion of the low-head/low-flow zone in the bottom-left corner prevents certain flow-head combinations from being realized, which may lead to oscillatory control in low-flow scenarios: For particular low head values, a larger flow may need to be selected than is desired to maintain a given water level, which may draw down the water level below target – requiring upwards compensation in a later time step. The risk of oscillatory

behaviour is another factor that has motivated the development of the invexity approach in Chapter 5.

The resulting optimization problem is the following:

$$
\begin{aligned}
&\min_{Q,H} \ 1 \\
&\text{s.t. } \underline{H}_i \le H_i(t_k) \le \overline{H}_i, && k = 1,\dots,T, \\
& && i = 1,\dots,m, \\
&\quad \frac{w_i l_i}{\Delta t}(H_i(t_k) - H_i(t_{k-1})) \\
&\qquad = Q_{i-1}(t_k) - Q_i(t_k), && k = 1,\dots,T, \\
& && i = 1,\dots,m, \\
&\quad 0 \le Q_i(t_k) \le \overline{Q}_i, && k = 1,\dots,T, \\
& && i = 1,\dots,m, \\
&\quad \underline{Q}_{w,i}(H_i(t_k)) \le Q_i(t_k) \le \overline{Q}_{w,i}(H_i(t_k)), && k = 1,\dots,T, \\
& && i = 1,\dots,m.
\end{aligned}
\tag{4.10}
$$

The lines demarcating the working area from the left and from the right are

$$
\underline{Q}_{w,i}(H_i(t_k)) = (H_i(t_k) - \overline{H}_{w,i})\overline{Q}_i^{1/3} C_{w,i}^{2/3}, \tag{4.11}
$$

$$
\overline{Q}_{w,i}(H_i(t_k)) = -\frac{1}{2}Q_{0,i} + C_{w,i}\frac{3}{2}\left(\frac{Q_{0,i}}{C_{w,i}}\right)^{1/3}(H_i(t_k) - \underline{H}_{w,i}), \tag{4.12}
$$

where $\underline{H}_w$ and $\overline{H}_w$ are the minimum and maximum possible crest levels, respectively, $C_{w,i}$ is the weir coefficient, and $Q_{0,i}$ is the nominal discharge (see Appendix 4.A).

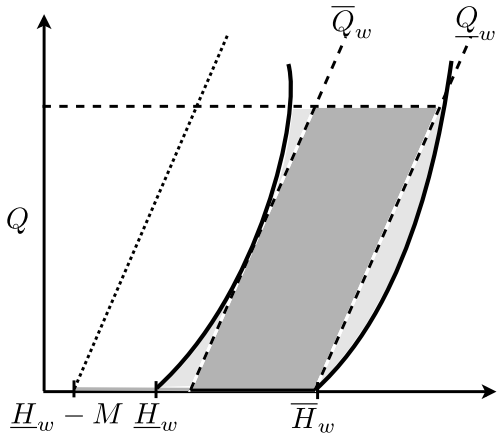


Figure 4.3 – Original working area (light gray) and approximated working area (dark gray) of a weir.

Mixed-integer modeling

The approach shown above does not take into account the case when the water level drops below the weir crest and the discharge is zero. This operation is equivalent to the horizontal dark-gray line in Figure 4.3, which should also be part of the working area. To make the operation possible in this region, the left constraint should be shifted more to the left and the discharge should be guaranteed to be zero if the weir is operating outside the parallelogram-shaped working area. Such a shape of working area leads to a mixed-integer convex optimization problem. Thus the boolean variables $\delta_i(t_k)$ are introduced, whose value is one when there is flow over the weir and zero when there is no flow, yielding

the following problem:

$$
\begin{aligned}
&\min_{Q,\delta,H} \ 1 \\
&\text{s.t. } \underline{H}_i \le H_i(t_k) \le \overline{H}_i, && k = 1,\dots,T, \\
& && i = 1,\dots,m, \\
&\quad \frac{w_i l_i}{\Delta t}(H_i(t_k) - H_i(t_{k-1})) \\
&\qquad = Q_{i-1}(t_k) - Q_i(t_k), && k = 1,\dots,T, \\
& && i = 1,\dots,m, \\
&\quad 0 \le Q_i(t_k) \le \delta_i(t_k)\overline{Q}_i, && k = 1,\dots,T, \\
& && i = 1,\dots,m, \\
&\quad \underline{Q}_{w,i}(H_i(t_k)) \le Q_i(t_k) \\
&\qquad \le \overline{Q}_{w,i}(H_i(t_k), \delta_i(t_k)), && k = 1,\dots,T, \\
& && i = 1,\dots,m, \\
&\quad \delta_i(t_k) \in \{0,1\}, && k = 1,\dots,T, \\
& && i = 1,\dots,m,
\end{aligned}
\tag{4.13}
$$

with $\underline{Q}_{w,i}$ as in Equation (4.11), and

$$
\begin{aligned}
&\overline{Q}_{w,i}(H_i(t_k), \delta_i(t_k)) = -\frac{1}{2}Q_{0,i} \\
&+ C_{w,i}\frac{3}{2}\left(\frac{Q_{0,i}}{C_{w,i}}\right)^{1/3}(H_i(t_k) - \underline{H}_{w,i} + M_i(1 - \delta_i(t_k))),
\end{aligned}
\tag{4.14}
$$

where $\delta_i(t_k)$ are boolean variables indicating whether there is flow over the weir, m is the number of the reaches, and M_i are sufficiently large constants such that the switching decisions are enforced correctly. The values of M_i can be understood as follows: they shift the left boundary of the working area by M_i in case there is no flow, *i.e.*, when $\delta = 0$, such that $Q = 0$ is feasible for all feasible values of H. This is shown by a dotted line in

Figure 4.3. In this case the working area is the gray parallelogram extended with a gray segment along the H axis starting at $\underline{H}_w - M$. In the practical example the bounds on the water level are implemented as soft constraints. The final formulation of the optimization problem is:

$$
\begin{aligned}
\min_{Q,\delta,H,\epsilon_1,\epsilon_2} \quad & \sum_{\substack{i=1,\dots,m,\\ k=1,\dots,T}} \left[\epsilon_{1,i}(t_k)^2 + \epsilon_{2,i}(t_k)^2\right] \\
\text{s.t. } & \underline{H}_i - \epsilon_{1,i}(t_k) \le H_i(t_k) \\
& \quad \le \overline{H}_i + \epsilon_{2,i}(t_k), && k = 1,\dots,T, \\
& && i = 1,\dots,m, \\
& \frac{w_i l_i}{\Delta t}(H_i(t_k) - H_i(t_{k-1})) \\
& \quad = Q_{i-1}(t_k) - Q_i(t_k), && k = 1,\dots,T, \\
& && i = 1,\dots,m, \\
& 0 \le Q_i(t_k) \le \delta_i(t_k)\overline{Q}_i, && k = 1,\dots,T, \\
& && i = 1,\dots,m, \\
& \underline{Q}_{w,i}(H_i(t_k)) \le Q_i(t_k) \\
& \quad \le \overline{Q}_{w,i}(H_i(t_k), \delta_i(t_k)), && k = 1,\dots,T, \\
& && i = 1,\dots,m, \\
& \delta_i(t_k) \in \{0,1\}, && k = 1,\dots,T, \\
& && i = 1,\dots,m, \\
& \epsilon_{1,i}(t_k) \ge 0, && k = 1,\dots,T, \\
& && i = 1,\dots,m, \\
& \epsilon_{2,i}(t_k) \ge 0, && k = 1,\dots,T, \\
& && i = 1,\dots,m,
\end{aligned}
\tag{4.15}
$$

with $\underline{Q}_{w,i}$ as in Equation (4.11), $\overline{Q}_{w,i}$ as in Equation (4.14), and with slack variables $\epsilon_{1,i}(t_k)$. If the optimal value of the slack variables is zero, then the water level H will stay within the bounds

according to the first constraint. The squares of the slack variables are minimized, as in this way, larger water level deviations are penalized more heavily.

Problem (4.15) is a mixed-integer optimization problem with convex-quadratic objective and linear constraints. The number of optimization variables depends on the size of the system that is being modelled (number of weirs, number of reaches) as well as on the optimization horizon and the temporal discretization step size.

4.3 Case study

The river Linge is part of the drainage system in the South of the Netherlands (see Figure 4.4). The Upper Linge has 12 reaches divided by weirs, and the Lower Linge consists of a single long reach. The Linge is used to collect the water from the polders and to conduct it to the North Sea through the river Merwede. The water leaves the Linge by free flow or by pumping depending on the water level in the Linge and the Merwede river. This case study only considers the Upper Linge.

The characteristics of the weirs and the geometry of the system are shown in Table 4.1. The backwater area and the time delay are obtained from Bronkhorst (2010).

4.3.1 Setting

The following test was conducted: while constant 0.1 m^3/s discharge was flowing in and out of the river, an hour-long discharge wave of 4.5 m^3/s coming from upstream was fed to the system after 6 hours. Two systems were tested during two days using (1) the proposed mixed-integer optimization, and (2) continuous optimization without water-level-dependent constraints on weir

Figure 4.4 – Location of the river Linge (dark blue) in the Netherlands.

Table 4.1 – Geometry of the reaches is obtained from Bronkhorst (2010), levels are given related to NAP (Amsterdam Ordnance Datum).

Branch name	**Backwater area** (m^2)	c_{idz} (s)	**Min. crest level** (NAP+m)	**Max. crest level** (NAP+m)	**Weir width** (m)
1	41682	1763	8.1	9.2	6.0
2	26416	1493	8.4	9.0	6.0
3	47601	2199	7.4	8.84	6.0
4	43848	418	7.41	8.4	6.0
5	47712	1565	6.8	7.97	6.0
6	76457	825	6.24	6.81	6.0
7	270461	1146	5.51	6.01	5.94
8	55691	1520	4.8	5.72	5.94
9	99111	1124	3.72	4.58	6.0
10	436163	64	2.42	3.35	9.5
11	103840	1235	1.47	2.26	9.5
12	210146	1767	-	-	-

discharge. If the continuous approach produced a discharge violating the crest level constraints, the weir discharge was snapped to the closest feasible value. The control time step was 1h, and the prediction horizon was 8h. MPC was implemented and tested in closed loop with SOBEK (Deltares 2016), a software package that solves the full Saint-Venant and weir equations. For the test case lexicographic goal programming was used (cf. Chapter 2). The first goal was to keep the water levels in the bounds shown in Table 4.1. In order to keep the water levels closer to the middle of the range, a second goal was added, where the bounds are 7 cm tighter on both sides.

The mixed-integer optimization problems had 176 continuous variables, and 88 binary variables each. They were solved using CPLEX wrapped in RTC-TOOLS 2 (Baayen et al. 2018).

4.3.2 Results and discussion

The results are shown in Figures 4.5-4.7. Each figure shows the controlled water level and the minimum and maximum bounds of water level (with a dashed line). Figure 4.5 shows how the water level increases suddenly due to the discharge wave. It can be seen how the wave travels through the river by the change of the water level in the reaches. It can also be seen that the water level is lowered before the wave is coming in order to accommodate the water volume as a result of the predictive capability of the control system.

The weir dynamics are compared in Figures 4.8-4.9. It can be seen that the weir dynamics of the continuous approach (red line) are different from those of the mixed-integer approach (black line). The mixed-integer controller lowers the weirs more before the discharge wave comes, to make more space for the incoming water.

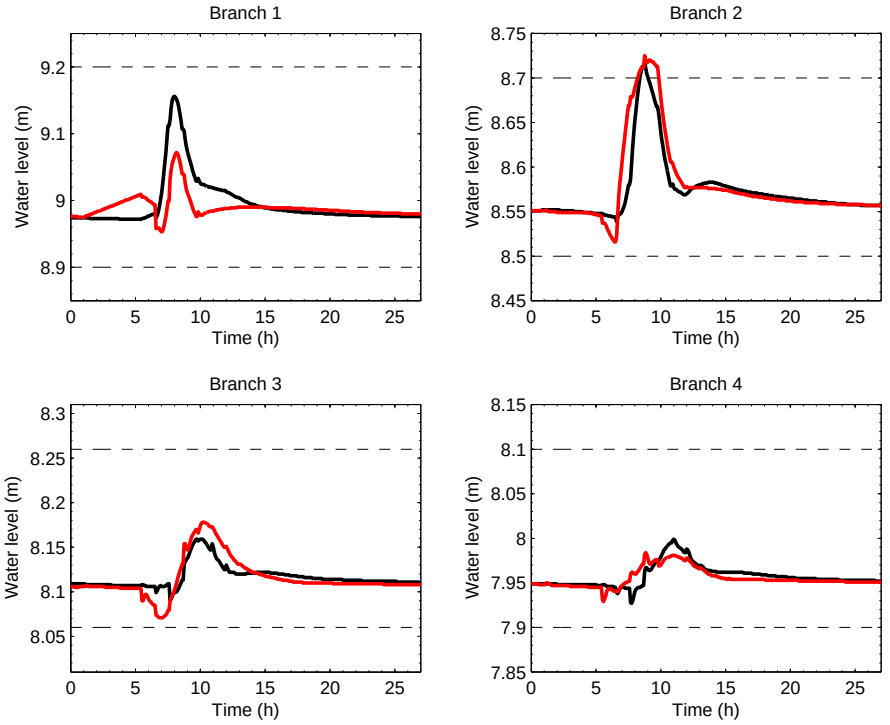


Figure 4.5 – Results of mixed-integer (red) and continuous (black) model, reaches 1-4.

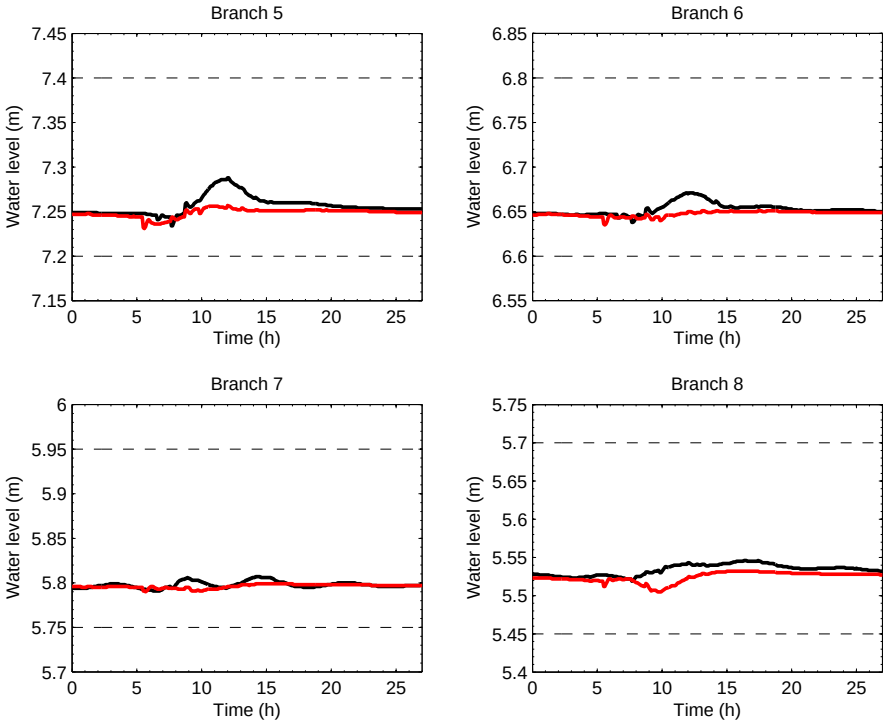


Figure 4.6 – Results of mixed-integer (red) and continuous (black) model, reaches 5-8.

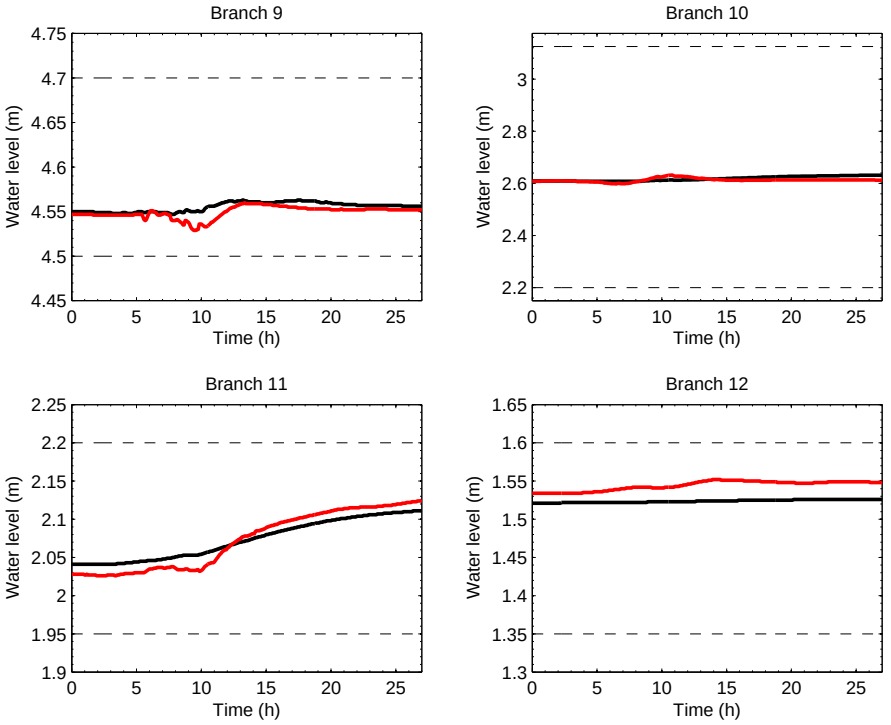


Figure 4.7 – Results of mixed-integer (red) and continuous (black) model, reaches 9-12.

This does not occur with the continuous controller: as the weir dynamics have no limit, the optimizer considers it sufficient to start lowering the weirs later. Is is therefore clear that the "knowledge" of the controller about the physical limitation of the weirs plays a crucial role in keeping the water levels within the bounds. These are the physical limitations, shown in Figure 4.3. The mixed-integer controller will only choose a weir discharge that is physically possible, and hence this chosen discharge will be equal to the one occurring at reality (in the SOBEK model). In case of the continuous approach, on the other hand, if the chosen discharge cannot be achieved due to the physical limitations of the weir, then the discharge chosen by the controller will not be equal to the discharge in reality.

4.4 Conclusions

A mixed-integer convex approximation was presented to the weir optimization problem. This approach was used to implement MPC for a river section with weirs. The MPC controller was implemented with the open source software package RTC-TOOLS 2, and was tested in a closed loop with a hydraulic simulation model. The controller was able to keep water levels within bounds using physically feasible weir crest levels. Finally, the mixed-integer approach resulted in a tighter tracking of the target water level than a continuous approximation approach.

4.4 Acknowledgments

This research was carried out within the *Slim Malen* (Smart Drainage) project. The Slim Malen project was funded and carried out by the following partners: STOWA, Ministry of Economic Affairs (RVO), Deltares, Eindhoven University of Technology, Min-

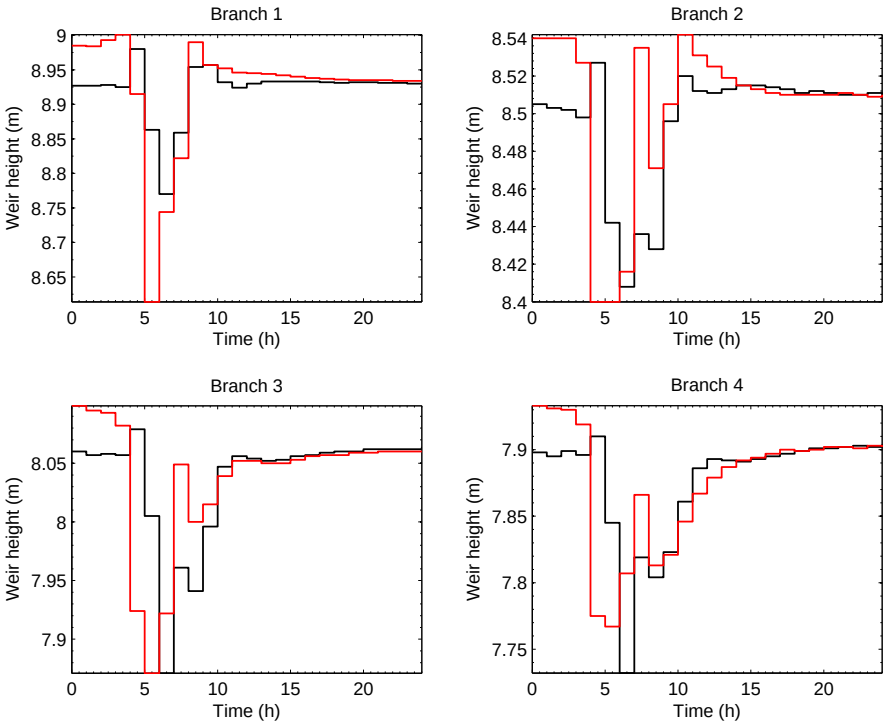


Figure 4.8 – Resulting crest levels, reaches 1-4. The continuous results are in red; the mixed-integer results are in black.

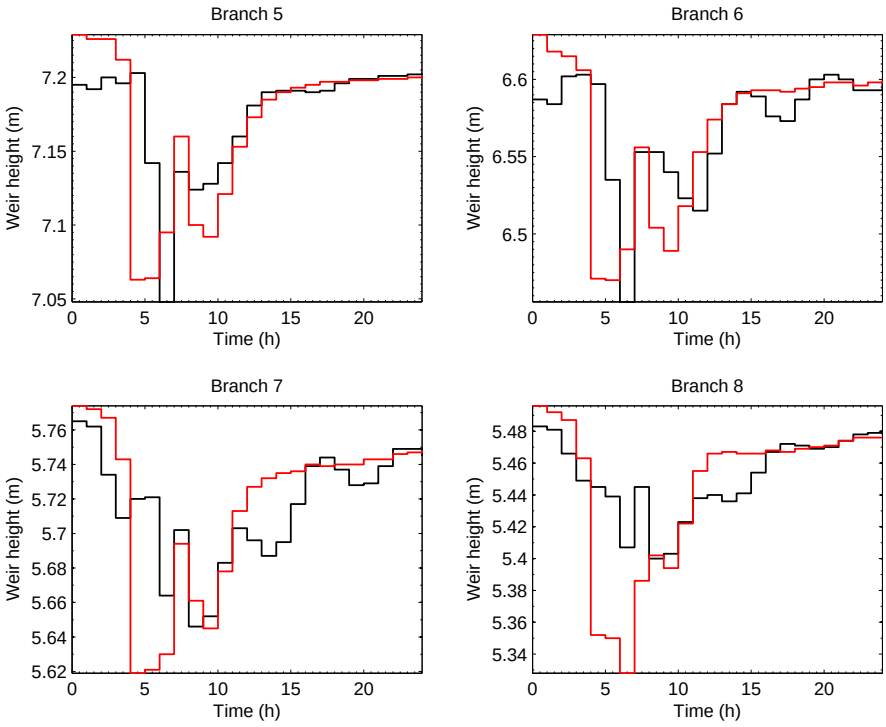


Figure 4.9 – Resulting crest levels, reaches 5-8. The continuous results are in red; the mixed-integer results are in black.

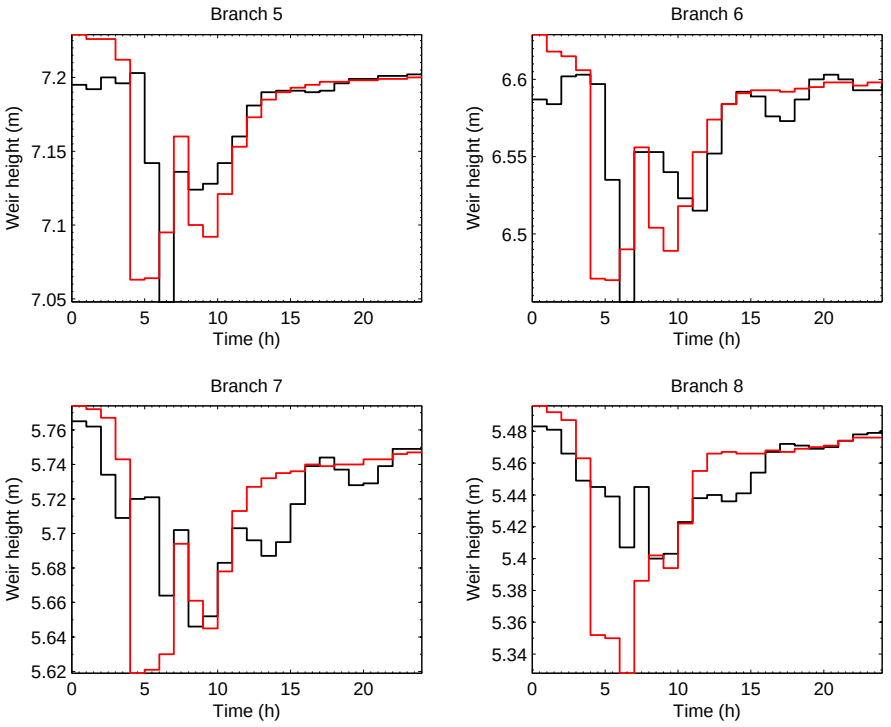


Figure 4.10 – Resulting crest levels, reaches 5-8. The continuous results are in red; the mixed-integer results are in black.

istry of Infra and Environment (Rijkswaterstaat – WVL), Water Authorities Hoogheemraadschap Hollands Noorderkwartier, Fryslân, Zuiderzeeland, Rivierenland, Scheldestromen, Rijnland, Brabantse Delta and Hollandse Delta and by companies Nelen & Schuurmans, e-Risk Group, Eneco, Delta, Alliander EXE, Actility, XYLEM and KISTERS.

Appendix

4.A Derivation of the boundaries of the working area

In this section we derive the equations describing the left and right boundaries of the weir working area.

4.A.1 Derivation of the maximum discharge boundary

The maximum discharge line, $\overline{Q}_w(H)$, is the linearization of the weir equation when $H = \underline{H}_w$ at the point $\overline{Q}/2$. Let us denote $Q_0 := \overline{Q}/2$, the nominal discharge. In case the most common operating discharge of the weir is known, this value can be chosen as nominal discharge, so that the linear approximation will be more accurate in the range where the weir is most often operated. The water level corresponding to this discharge is denoted H_0. The following weir equation holds:

$$Q_0 = C_w(H_0 - \underline{H}_w)^{3/2}. \tag{4.16}$$

Rearranging this equation the nominal water level (around which we linearize), H_0 can be calculated:

$$H_0 = \left(\frac{Q_0}{C_w}\right)^{2/3} + \underline{H}_w. \tag{4.17}$$

By differentiating (4.16) to H_0, the slope of the line $\overline{Q}_w(H)$ can be calculated:

$$S = C_w \frac{3}{2}\sqrt{H_0 - \underline{H}_w}. \tag{4.18}$$

The equation of the tangent line $\overline{Q}_w(H)$ to the curve at the point (Q_0, H_0) can be written as:

$$\overline{Q}_w = S(H - H_0) + Q_0.$$

In order to be able to shift this line to the left by M in case of no flow, the boolean δ is introduced, the value of which is one if there is flow over the weir and zero otherwise. The previous equation can be written as:

$$\overline{Q}_w = S(H - H_0 + M(1-\delta)) + Q_0.$$

Substituting the slope (4.18) into the above equation yields the following:

$$\overline{Q}_w = C_w \frac{3}{2}\sqrt{H_0 - \underline{H}_w}(H - H_0 + M(1-\delta)) + Q_0.$$

Also the nominal water level (4.17) can be substituted:

$$\overline{Q}_w = C_w \frac{3}{2}\left(\frac{Q_0}{C_w}\right)^{1/3}\left(H - \left(\frac{Q_0}{C_w}\right)^{2/3} - \underline{H}_w + M(1-\delta)\right) + Q_0.$$

Rearranging this equation, the following equation is derived for the maximum discharge bound corresponding to the minimum crest level:

$$\overline{Q}_w = -\frac{1}{2}Q_0 + C_w \frac{3}{2}\left(\frac{Q_0}{C_w}\right)^{1/3}(H - \underline{H}_w + M(1-\delta)).$$

4.A.2 Derivation of the minimum discharge boundary

The minimum discharge boundary is a line $\underline{Q}_w$ that connects two points: ($\overline{H}_w$,0) and $((\overline{Q}/C_w)^{2/3} + \overline{H}_w, \overline{Q})$. Let us write the equation for this line in the following form:

$$\underline{Q}_w = a(H - \overline{H}_w) + b. \tag{4.19}$$

By inserting the first point into the equation of the line, (4.19), we obtain $b = 0$. The slope of the line is obtained by inserting the second point as:

$$a = \frac{\overline{Q}}{(\overline{Q}/C_w)^{2/3}}.$$

Finally, inserting a and b into (4.19), an expression for $\underline{Q}_w$ is obtained:

$$\underline{Q}_w = \frac{\overline{Q}}{(\overline{Q}/C_w)^{2/3}}(H - \overline{H}_w).$$

Rearranging the above equation, the final expression for $\underline{Q}_w$ is:

$$\underline{Q}_w = (H - \overline{H}_w)\overline{Q}^{1/3}C_w^{2/3}.$$

Part II

Invex optimization of flow network problems

CHAPTER 5

Hidden invariant convexity for global and conic-intersection optimality guarantees in discrete-time optimal control

5.1 Introduction

Discrete-time optimal control typically involves the solution of an optimization problem, which need not be convex. Such non-convex optimization problems arise, for example, in systems driven by nonlinear partial differential equations (PDEs), such as water, gas, and power systems (Ackermann et al. 2000, Burgschweiger et al. 2009, Hante et al. 2017). A typical objective in such problems is to steer the system into tracking target values for certain state variables, *e.g.*, stabilizing a water level around the desired level (García et al. 1989).

One of the desiderata for a solution to an optimization problem is *global optimality*. Although locally optimal solutions are often used, globally optimal ones typically yield substantially better objective values. While approaches aiming for global optimality in general nonlinear optimization have been proposed (Sahinidis

1996, Belotti et al. 2009, Misener and Floudas 2014, Ghaddar et al. 2017), these approaches are often incompatible with tight computation time limits or large problem sizes. Therefore, common work-arounds are to use linearizations (Eschenbach et al. 2001, Amann et al. 2016, Falk et al. 2016), convex restrictions, or convex relaxations (Madani et al. 2014, Horváth et al. 2019, Lee et al. 2019), which provide tractability yet at the cost of model accuracy, or to resort to genetic algorithms (Van Zyl et al. 2004, Nicklow et al. 2010, Vermuyten et al. 2018).

It is most desirable, however, to obtain a globally optimal solution to the "most exact" nonlinear model without resorting to computationally expensive techniques. Ample numerical evidence exists that locally optimal solutions to non-convex discrete-time optimal control problems are often of high quality, hardly distinguishable from true global optima and/or better than other applied methods (Burgschweiger et al. 2009, Joseph-Duran et al. 2014, Ghaddar et al. 2017, Baayen et al. 2019b).

In this chapter, we provide a theoretical underpinning for this phenomenon by showing that, in the problems we study, the objective function composed with the dynamics is *invex* in the original sense of Hanson (Hanson 1981) and Craven (Craven 1981b). Invexity is a generalization of convexity that certifies, roughly speaking, that on an open set a stationary point of an invex function is its global minimum (Ben-Israel and Mond 1986, Mishra and Giorgi 2008). Despite the large body of theoretical work, we are not aware of research leveraging invexity in large-scale applied (engineering) context.

We use invexity to prove optimality guarantees for KKT points of problems belonging to a class of discrete-time optimal control problems, including full global optimality for KKT points in the interior of the feasible set. Because the invexity of the involved

functions is not readily seen and arises from a function implicitly defined by the problem's constraints, we refer to it as *hidden invexity*. This is analogous to the term *hidden convexity* (Ben-Tal and Teboulle 1996) that is used when the convexity of a problem is not immediately apparent. The hidden convexity result of Ben-Tal and Teboulle (1996) showed that by reformulating an original, non-convex problem, one can solve a convex problem whose optimal solution is a global optimum of the non-convex problem.

Our result, in turn, relies on a direct analysis of the non-convex problem using the notion of invexity and the language of tangent cones. It allows us to prove global optimality properties for KKT points directly. This is also how our result differs from convex restriction and convex relaxation methods.

The research contribution of our work, seen from the perspective of different fields, is as follows:

(a) From the mathematical optimization angle, we show that for a large class of non-convex problems, certifying the hidden invexity and using standard local solvers is a viable alternative to the use of solvers for general non-convex optimization problems.

(b) From the optimal control angle, we show that for a large class of discrete-time optimal control problems, invex formulations exist, preserving the exact nonlinear dynamics. This allows to tractably determine high-quality solutions to large-scale problems.

(c) From the nonlinear analysis angle, we show that the hidden invexity of a discrete-time optimal control problem yields so-called *conic-intersection* optimality guarantees for KKT points with active inequality constraints, and global optimality for KKT points in the interior of the domain of the

control variables. The notion of conic-intersection optimality will be formally defined in Section 5.3.

The remainder of this chapter is structured as follows. Section 5.2 introduces the notion of *regular* problems for which we establish our result. In Section 5.3 we prove the main result of the chapter. Section 5.4 presents a numerical study for a single river segment modelled using a nonlinear PDE.

5.2 Regular discrete-time optimal control problems

In this section we describe the class of discrete-time optimal control problems for which we demonstrate invexity. Consider the problem

$$\begin{aligned} \min_{x,u} \ & (f \circ g)(x) \\ \text{s.t. } & c(x,u) = 0 \\ & d(u) \leq 0, \end{aligned} \tag{$\mathcal{P}$}$$

where we refer to the variables $x \in \mathbb{R}^m$ as *states* and the variables $u \in \mathbb{R}^n$ as *controls*. These names are motivated by the fact that the controls *implicitly* determine the values of the states through the *equality constraints* $c(x,u) = 0$. The function $f : \mathbb{R}^n \to \mathbb{R}$ is the *objective* and the function $g : \mathbb{R}^m \to \mathbb{R}^n$ is the *output function* mapping states x to *outputs* $y := g(x)$. These concepts originate in control theory and are essential to our analysis. The relationship between the controls u, the implicitly defined states x, and the output variables y, is illustrated in Figure 5.1.

We denote the set of admissible controls as $U := \{u \in \mathbb{R}^n : \ d(u) \leq 0\}$, where d is a vector-valued function whose components are the *control inequality constraint functions*, with the inequality holding

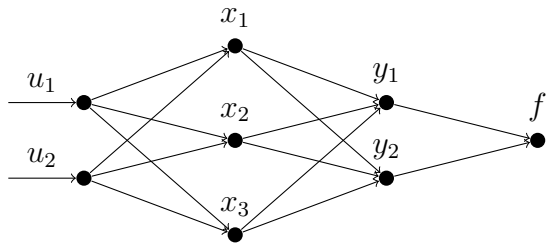


Figure 5.1 – The relationship between the controls u, the implicitly defined states x, and the output variables y, for $n = 2$ and $m = 3$.

component-wise, and denote the set of indices of the coordinates of d by $\mathcal{I}$.

Our goal is to show that the objective of problem $(\mathcal{P})$ is invex as a function of the controls u, under certain conditions. However, the equality constraints in $(\mathcal{P})$ can involve nonlinear functions, making the analysis cumbersome. We shall alleviate this difficulty by eliminating the constraints using implicit function theory and analyzing the problem using *total* gradients with respect to u, wherein the derivatives of the state variables x_i with respect to u are expressed explicitly. This step is used for the analysis, but is not required in numerical implementation.

If the Jacobian $\nabla_x c$ is invertible, then the total Jacobian of the states x with respect to the controls u may be expressed using the implicit function theorem as

$$D_u x = -\nabla_x^{-1} c \nabla_u c, \tag{5.1}$$

in which the prefix ∇_x denotes the matrix of partial derivatives with respect to the components of x, and D_x the matrix of total derivatives with respect to x, of a given function. In what follows, by ∇^{-1} and D^{-1} we denote the inverses of the respective matrices, if they exist.

In order to establish our result, we need some conditions. In the

following, we define the *regular* problems for which invexity can be demonstrated. After the definition, we discuss each of the conditions.

Definition 5.1. *Consider problem* $(\mathcal{P})$ *with states* x *and controls* u*. Let the functions* f*,* g*,* c *and* d *be continuously differentiable. We say that* $(\mathcal{P})$ *is* regular *if the following conditions are satisfied:*

the linear independence constraint qualifications *(LICQ):*

1. *the Jacobian matrix of the equality constraints* c *with respect to the state vector* x*,* i.e.*,* $\nabla_x c(x,u)$*, is square and full-rank for all* (x,u) *such that* $c(x,u)=0$*,* $u \in U$*,*

2. *the gradient vectors of the active inequality constraints* d_i *at the point* u*,* i.e.*,* $\nabla_u d_i(u)$ *for all* $i \in \mathcal{I}$ *such that* $d_i(u)=0$*, are linearly independent for all* $u \in U$*,*

the uniqueness condition*:*

3. *for all* $u \in U$*, the constraints* $c(x,u)=0$ *have a unique solution* x*,*

the output-controllability condition*:*

4. *the output function* $g : \mathbb{R}^m \to \mathbb{R}^n$ *is such that the square matrix*
$$-\nabla_x g(x)\, \nabla_x^{-1} c(x,u)\, \nabla_u c(x,u), \tag{5.2}$$
is invertible for all (x,u) *such that* $c(x,u)=0$*,* $u \in U$*,*

and the convexity condition*:*

5. *the objective function* $f : \mathbb{R}^n \to \mathbb{R}$ *is convex.*

Note that in problem $(\mathcal{P})$, no explicit bounds or inequality constraints are imposed on the state vector x. State bounds $e(x) \leq 0$ do not fit the formalism $d(u) \leq 0$ of problem $(\mathcal{P})$ whence, in general, a problem with explicit state bounds is not regular. In practice, imposing explicit state bounds can lead to a situation where the projection of the feasible set in terms of (u, x) onto the space of u becomes a disconnected set. However, terms penalizing deviation from output variable "bounds" may be included in the objective function f.

We now discuss each of the conditions of Definition 5.1.

Condition 1 is a linear independence constraint qualification (LICQ, Nocedal and Wright (2006)). It is required for the state vector x to be well-defined as an implicit function of u, and for us to be able to apply the implicit function theorem. Condition 1 is often straightforward to demonstrate if the dynamics of the underlying model are integrable in time. For the dynamics to be integrable in time, it is required that the number of states be equal to the number of equations, and furthermore that the Jacobian of the equations with respect to the states is non-singular.

Condition 2 is the second LICQ and is satisfied for standard domains such as balls and boxes. It prevents the constraints $d(u) \leq 0$ from defining a lower-dimensional subspace of $\mathbb{R}^n$.

Condition 3 states that the implicit function $u \mapsto x$ is uniquely defined on U. It is a standard assumption in discrete-time optimal control, as this required uniqueness property typically is satisfied if the dynamics are (uniquely) integrable in time. Note that by Condition 1 alone, implicit function theory would only provide for an implicit function that is *locally* unique. It would not guarantee an implicit function that is uniquely defined on all of U.

Condition 4 states that different states should map to different

outputs. Conditions 1 and 2 imply that the implicit function $u \mapsto x$ is injective. Therefore $x[U]$, *i.e.*, the image of the set of admissible controls under the implicit function x, is an n-dimensional manifold in $\mathbb{R}^m$. Any mapping g that is invertible on this manifold $x[U]$ is therefore a valid choice.

Another interpretation of Condition 4 is the following. If we would require the LICQ and uniqueness conditions to hold on $\mathbb{R}^n$ (rather than only on U), Condition 4 would imply that for every $y \in \mathbb{R}^n$, there would exist a control input $u \in \mathbb{R}^n$ such that $y = (g \circ x)(u)$. This hypothetical condition is exactly the classical *output-controllability* (Ogata and Yang 1997) condition.

Condition 5 is standard and includes objectives such as p-norms raised to the p-th power with $p \geq 2$.

Before looking at examples of regular problems, it is instructive to consider a few irregular problems to show that the regularity conditions indeed eliminate some of the well-known NP-hard problems.

Example 5.2.1. *Let $[0, 1] \subset U \subset \mathbb{R}$. If problem $(\mathcal{P})$ contains a binary restriction constraint $u(1-u) = 0$, then Condition 4 is not satisfied at $u = 0.5$.*

Example 5.2.2. *Let $U \subset \mathbb{R}$. If problem $(\mathcal{P})$ contains binary restriction constraints of the form*

$$\begin{aligned} -u &\leq 0, \\ u &\leq 1, \\ u(1-u) &\leq 0, \end{aligned}$$

then Condition 2 is not satisfied at $u = 0$ and at $u = 1$.

Example 5.2.3. *Let $[-1, 1] \subset U \subset \mathbb{R}$. If problem $(\mathcal{P})$ contains a sinusoidal constraint $u = \sin x$, then Condition 3 is, in general, not satisfied.*

Example 5.2.4. *Let $0 \in U \subset \mathbb{R}$. If problem $(\mathcal{P})$ contains a bilinear constraint of the form $u = x_1 x_2$, then the LICQ Condition 1 is, in general, not satisfied.*

Example 5.2.5. *Let $(0,0) \in U \subset \mathbb{R}^2$. If problem $(\mathcal{P})$ contains a bilinear constraint of the form $x = u_1 u_2$, then Condition 4 is, in general, not satisfied.*

Example 5.2.6. *Let $0 \in U \subset \mathbb{R}$. If problem $(\mathcal{P})$ contains a piecewise constraint of the form*

$$x = \begin{cases} 0 & \text{if } u_1 < 0, \\ u_1^2 & \text{otherwise,} \end{cases}$$

then Condition 4 is, in general, not satisfied whenever $u_1 < 0$.

We now proceed to introduce examples of systems meeting the regularity conditions. A first example is provided by affine constraint functions satisfying the appropriate rank conditions.

Example 5.2.7. *Consider problem $(\mathcal{P})$ with control vector $u \in \mathbb{R}^n$, state vector $x \in \mathbb{R}^m$, output vector $y \in \mathbb{R}^n$, and trajectory tracking objective*

$$f(y) = \sum_{i=1}^{n} |y_i - y_i^t|^p$$

with $p \in [2, \infty)$ and affine output function

$$y = Cx + c,$$

subject to the bounds

$$-\infty < u_j^L \leq u_j \leq u_j^U < \infty \quad j \in \{1, \ldots, n\},$$

and affine constraints

$$c(x, u) = Ax + Bu + b,$$

with matrix A square and invertible, matrices B and C full rank and matrix C such that the square matrix $CA^{-1}B$ is invertible. This problem is regular.

Our next example considers trigonometric constraints, which commonly arise in control of systems with axes of rotation such as vehicles, ships, and aircraft.

Example 5.2.8. *Problem $(\mathcal{P})$ with control vector $u \in \mathbb{R}^2$, state vector $x \in \mathbb{R}^2$, output vector $y \in \mathbb{R}^2$, and trajectory tracking objective*

$$f(y) = \sum_{i \in \{1,2\}} |y_i - y_i^t|^p$$

with $p \in [2, \infty)$ and output function

$$y = x,$$

subject to the bounds

$$0 < u_1^L \leq u_1 \leq u_1^U < \infty,$$
$$0 \leq u_2 < 2\pi,$$

and constraints

$$x_1 = u_1 \cos u_2,$$
$$x_2 = u_1 \sin u_2,$$

is regular. Note that for Condition 4 to hold, u_1 must be bounded away from zero.

Some problems with piecewise-defined constraints that are also continuously differentiable, are regular:

Example 5.2.9. *Problem* $(\mathcal{P})$ *with control vector* $u \in \mathbb{R}^2$*, state vector* $x \in \mathbb{R}^3$*, output vector* $y \in \mathbb{R}^2$*, and trajectory tracking objective*

$$f(y) = \sum_{i \in \{1,2\}} |y_i - y_i^t|^p$$

with $p \in [2, \infty)$ *and output function*

$$y_1 = x_1,$$
$$y_2 = x_3,$$

subject to the bounds

$$u^L \leq u \leq u^U,$$

and constraints

$$x_1 = u_1,$$
$$x_2 = \begin{cases} 0 & \text{if } x_1 < 0, \\ x_1^2 & \text{otherwise,} \end{cases}$$
$$x_3 = u_2 + x_2,$$

is regular. Unlike in Example 5.2.6, the state vector x *of this example is always sensitive to the control variables* u_1 *and* u_2*.*

Problems with structure similar to that of Example 5.2.9 commonly occur when modelling water flow over a weir (a small dam), the crest of which may or may not lie below the upstream water surface.

Since bilinear constraints are very common, we also show how certain bilinear problems satisfy the regularity conditions.

Example 5.2.10. *Problem* $(\mathcal{P})$ *with control vector* $u \in \mathbb{R}^2$*, state vectors* $x \in \mathbb{R}^2$ *and* $z \in \mathbb{R}^2$*, output vector* $y \in \mathbb{R}^2$*, and objective*

$$f(y) = \sum_{i \in \{1,2\}} |y_i - y_i^t|^p$$

with $p \in [2, \infty)$ *and output function*

$$y = z,$$

subject to the bounds

$$0 \leq u_j^L \leq u_j \leq u_j^U < \infty \quad j \in \{1, 2\},$$

and constraints

$$u_1 = x_1 z_0, \tag{5.3}$$
$$x_1 = z_0 - z_1, \tag{5.4}$$
$$u_2 = x_2 z_1, \tag{5.5}$$
$$x_2 = z_1 - z_2, \tag{5.6}$$

with the fixed initial condition $z_0 \in \mathbb{R}$*, is regular as long as* z_0 *is chosen such that* $z_i \neq 0$*,* $i \in \{1, 2\}$*, for all feasible* u*. Conditions 1–2 and 4–5 are readily verified. To verify Condition 3,* i.e.*, that for any* $u \in U$ *the constraints admit a unique solution, note that the constraints* (5.3)–(5.6) *may be solved in the given order, starting from the fixed initial value* z_0*, inserting the computed value for* x_1 *into the subsequent equation, etc.*

Problems with structure similar to that of Example 5.2.10 commonly occur when modelling the power generation of a hydroelectric turbine in a power station. Instantaneous generation (u) is non-negative and bounded, and it is bilinear in flow (x) and the water level difference (z) across a dam, which is never zero. At the same time, an increase in flow (x) results in a decrease of the water level difference (z). Similar reasoning applies to the power consumption of pumps.

The conditions of Definition 5.1 are satisfied by certain optimization problems constrained by discretized hyperbolic PDEs, if a

suitable discretization is chosen. An example of an optimization problem constrained by appropriately discretized Saint-Venant equations (Vreugdenhil 2013) is presented and analyzed in Section 5.4.

In the next Section, we present our main theoretical result.

5.3 Hidden invexity

5.3.1 Introduction

In this section we present our main result that regular problems, in the sense of Definition 5.1, have *hidden invexity* when reduced to optimization over control variables. We first give the definition of invexity.

Definition 5.2. *Consider an open set $X \subset \mathbb{R}^n$. A function $h : X \to \mathbb{R}$ is called* invex *on X if there exists a vector function $\eta_h(x_2, x_1) : X \times X \to \mathbb{R}^n$ such that*

$$h(x_2) - h(x_1) \geq \eta_h^T(x_2, x_1)\nabla h(x_1), \tag{5.7}$$

for all $x_1, x_2 \in X$.

Note that the above becomes a definition of convexity in case $\eta_h(x_2, x_1) = x_2 - x_1$.

The name invex follows from *invariant convex* (Craven 1981b). A function is invex if and only if every stationary point is a global minimum. To see the first implication, set $\nabla h = 0$ in (5.7). A concise proof of the reverse implication can be found in (Ben-Israel and Mond 1986, Theorem 1).

The definition of invexity is usually stated for functions defined on open sets, wheareas our goal is to optimize over a closed set U defined by inequality constraints. There exists an entire family

of extensions of the notion of invexity to constrained optimization problems (KT-invexity (Martin 1985), HC-invexity (Hanson 1981, Craven 1981b, Martin 1985), Type I/Type II invexity (Hanson 1999)). However, each of them is difficult to apply to real-world problems like ours, due to the need to find a common function η for the objective and the constraints. Therefore, we shall stay with the standard notion of invexity and eliminate the equality constraints from the problem in order to show invexity of the objective function on the interior of U. In this process, we extend the analysis to the boundary of the feasible set U using the geometry of the tangent cones.

5.3.2 Main result

In our analysis, instead of constructing the function η_h of Definition 5.2 explicitly, we will use the fact that invexity of functions arises naturally in the composition of convex functions with transformations that are full-rank, *i.e.*, that have an invertible Jacobian (Craven 1981a). We will now show how regular problems fit this scheme.

In the following we assume that all conditions of Definition 5.1 hold. Per Condition 5 of Definition 5.1, the objective function f of a regular problem $(\mathcal{P})$ is convex. In order to obtain an invex function on the set of admissible controls U, we compose the objective f with a full-rank transformation. We construct this full-rank transformation by using the implicit function theorem to express the state vector x as a function $u \mapsto x$. Condition 1 ensures that the conditions of the implicit function theorem are met, and Condition 3 ensures that the function $u \mapsto x$ is globally

unique. Problem $(\mathcal{P})$ can therefore be rewritten as:

$$\begin{aligned} &\min_{u} \ (f \circ g \circ x)(u) \\ &\text{s.t. } d(u) \leq 0. \end{aligned} \tag{$\mathcal{P}^U$}$$

In $(\mathcal{P}^U)$, the composition $T : U \to Y := T[U]$, $T(u) := (g \circ x)(u)$, will be playing the role of the invertible transformation, and the composition $f \circ T$ will be shown to be invex. This setup is illustrated in Figures 5.1 and 5.2.

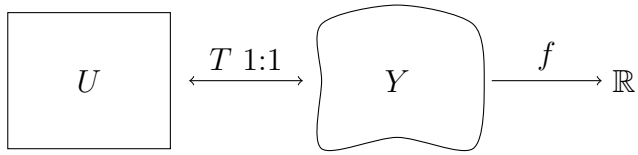


Figure 5.2 – Convex objective function f composed with invertible transformation T.

The strength of our result for a particular KKT point with (u^*, λ^*) will depend on the place where point $y^* = T(u^*)$ is in the set Y: in the interior, or on the boundary. In general, the set Y is non-convex and optimality guarantees are stated in terms of "the set of points in Y that can be seen" from y^*. To make this rigorous, we first recall the definition of the tangent cone (Geiger and Kanzow 2013, Nocedal and Wright 2006).

Definition 5.3. *Let $Y \subset \mathbb{R}^n$ be a non-empty set. A vector $d \in \mathbb{R}^n$ is called* tangent *to Y at $y \in Y$, if there exist sequences $\{y^k\} \subset Y$, $\{t_k\} \subset \mathbb{R}^+$ such that*

$$y^k \to y, \quad t_k \to 0, \quad \frac{y^k - y}{t_k} \to d.$$

The set of all tangent vectors at $y \in Y$ is the tangent cone *of Y at y, denoted as $\mathcal{T}_Y(y)$.*

Tangent cones provide the vocabulary to define our notion of *conic-intersection minimum.*

Definition 5.4. *Consider a regular problem* $(\mathcal{P}^U)$*. We say that a point* u^* *is a conic-intersection minimum if it is a global minimum on the set*

$$V(u^*) := \{u \in U : T(u) - T(u^*) \in \mathcal{T}_Y(T(u^*))\}, \tag{5.8}$$

where $\mathcal{T}_Y(T(u^*))$ *denotes the tangent cone of* Y *at* $T(u^*)$*.*

The definition states, in rough terms, that a conic-intersection minimum is a global minimum with respect to the interior of the domain and all inactive boundary segments (invexity), minus any points "hidden from view" due to local non-convexity of the active boundary segments in Y. The geometric meaning is illustrated in Figure 5.3.

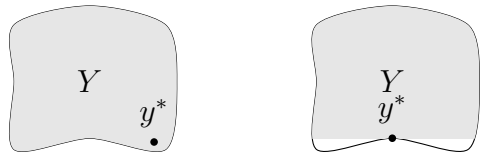


Figure 5.3 – The highlighted areas illustrate sets $T[V(u^*)]$ within which a solution $y^* = T(u^*)$ is globally optimal, for an interior (left) and a boundary solution (right). The sets are shown in the output space $Y = T[U]$ to highlight the role of the tangent cones. Note that in the left panel the point y^* is in the interior of Y, hence a global minimum in Y, despite the fact that there are some points in Y that are not "visible" from the tangent cone at y^*. This is thanks to invexity.

We now state our main result whose proof is presented in Section 5.3.3.

Theorem 5.1. *Consider a regular problem* $(\mathcal{P}^U)$*. Let* (u^*, λ^*) *be a KKT point of this problem. Then* u^* *is a conic-intersection minimum.*

Before proceeding to the corollary, we note that the reverse statement of Theorem 5.1, *i.e.*, that every minimum is a KKT point, follows from the LICQ Conditions 1 and 2 in Definition 5.1. These LICQ conditions form the regularity condition required for every minimum to be a KKT point (Nocedal and Wright 2006, Theorem 12.1).

The corollary is a direct consequence of the fact that for an interior point, $\mathcal{T}_Y(T(u^*)) = \mathbb{R}^n$.

Corollary 5.1. *Consider a regular problem* $(\mathcal{P}^U)$*. Consider a KKT point* (u^*, λ^*) *such that* $u^* \in \operatorname{int} U$*. Then* u^* *is a global minimum of* $f \circ T$ *on* U*.*

Corollary 5.1 is exactly the characterization of an invex function on $\operatorname{int} U$. In other words, then, the function $h := f \circ T$ is invex on $\operatorname{int} U$, with

$$\eta_h(u_2, u_1) = \nabla^{-1} T(u_1) \left(T(u_2) - T(u_1)\right).$$

The expression for η_h follows from the convexity, hence invexity, of f on $Y = T[U]$ with $\eta_f(y_2, y_1) = y_2 - y_1 = T(u_2) - T(u_1)$ (cf. Definition 5.2). By continuity of h we also know that none of the points on the boundary of U map to lower objective values, whence a KKT point in the interior of U is a global minimum on all of U.

We will now explain the meaning of the general result. For this, it is instructive to first recall the reference situation: general nonlinear optimization. A KKT point of a nonlinear optimization problem need not be a local minimum; it may also be a local maximum, or a saddle point. Furthermore, in case that a KKT point is a local minimum, it is only guaranteed to be minimal within an arbitrarily small neighbourhood of itself. From a numerical point of view, generic nonlinear optimization problems are hard:

local solvers may converge to KKT points that are local maxima or saddle points.

For a regular problem, Theorem 5.1 provides a stronger characterization of KKT points. Although there is no guarantee that a KKT point (u^*, λ^*) on the boundary of U is not a saddle point or a local maximum with respect to U, Theorem 5.1 states that u^* is a local minimum, and therefore not a maximum or a saddle point, with respect to $V(u^*)$. This is important from a numerical point of view, since it implies that a local solver will converge to a local minimum (and certainly not a maximum) with respect to $V(u^*)$. Secondly, it states that the point u^* is in fact a global minimum over $V(u^*)$.

The meaning of the global optimality of the KKT point u^* over $V(u^*)$ is best understood by analyzing two distinct cases. The first is when $u^* \in \operatorname{int} U$, *i.e.*, u^* lies in the interior of U, in which case $V(u^*) = U$ so that u^* is a global minimum over U. The second is when u^* lies on the boundary of U. In that case, its objective value is no greater than the objective values for all points u corresponding to points y that lie in the intersection of the translated tangent cone with the set Y itself, *i.e.*, that lie in the set $(T(u^*) + \mathcal{T}_Y(T(u^*))) \cap Y$. Both cases are illustrated in Figure 5.3.

A conic-intersection minimum on the boundary need not be globally optimal, as is demonstrated by the following example:

Example 5.3.1. *Problem* $(\mathcal{P})$ *with control vector* $u \in \mathbb{R}^2$*, state vector* $x \in \mathbb{R}^2$*, output vector* $y \in \mathbb{R}^2$*, and objective*

$$f(y) = -y_2,$$

and output function

$$y = x,$$

subject to the bounds

$$\begin{aligned} -0.5 \leq u_1 \leq 1, \\ 0 \leq u_2 \leq 1, \end{aligned}$$

and constraints

$$\begin{aligned} x_1 &= u_1, \\ x_2 &= u_2 + u_1^2, \end{aligned}$$

is regular. The point $u^* = (-0.5, 1)$ *is a KKT point on the boundary, and therefore a conic-intersection minimum by Theorem 5.1. It is thus a global minimum on the set*

$$\begin{aligned} V(u^*) = \{(u_1, u_2) : -0.5 \leq u_1 \leq 1,\ 0 \leq u_2 \leq 1, \\ u_2 + u_1^2 \leq 1.25 - (u_1 + 0.5)\}. \end{aligned}$$

The point u^* *is not, however, globally optimal on* U*, since a lower objective function value is obtained at the point* $(1, 1)$*.*

A real-world application of Theorem 5.1 is discussed in Section 5.4.

5.3.3 Proof of Theorem 5.1

Recall that the function $u \mapsto x$ exists by virtue of the implicit function theorem, the conditions of which are satisfied due to Condition 1 of Definition 5.1. By Condition 3, this function is uniquely defined on the feasible set U.

Consider the transformation $T = g \circ x$. By Condition 4 of Definition 5.1, the matrix $D_u T$ is invertible whence, by the inverse function theorem, the transformation T itself is invertible. The transformation and its use within the optimization problem is illustrated in Figure 5.2.

As per the definition of problem $(\mathcal{P})$, the feasible set is described using the constraints $d(u) \leq 0$. We will first show that a point (u^*, λ^*) is a KKT point of the optimization problem

$$\begin{aligned} &\min_u \ (f \circ T)(u) && (\mathcal{P}^U) \\ &\text{s.t. } d(u) \leq 0, \end{aligned}$$

if and only if $(T(u^*), \lambda^*)$ is a KKT point of the optimization problem

$$\begin{aligned} &\min_y \ f(y) && (\mathcal{P}^Y) \\ &\text{s.t. } (d \circ T^{-1})(y) \leq 0. \end{aligned}$$

Afterwards, we will analyze the global optimality structure of the KKT points. Let

$$\mathcal{L}^U(u, \lambda) := (f \circ T)(u) + \lambda^T d(u)$$

denote the Lagrangian of problem $(\mathcal{P}^U)$, and let

$$\mathcal{L}^Y(y, \lambda) := f(y) + \lambda^T (d \circ T^{-1})(y)$$

denote the Lagrangian of problem $(\mathcal{P}^Y)$. We will use the standard definition of KKT points following Nocedal and Wright (2006). KKT points (u, λ) of $(\mathcal{P}^U)$ correspond to stationary points (u, λ) of the Lagrangian $\mathcal{L}^U$ by which we have, using the fact that the transformation T is invertible, that:

$$\begin{aligned} 0 &= D_y \mathcal{L}^Y \\ &= \nabla_y f + \lambda^T \nabla_u d \, D_u \left(T^{-1} \right) \\ &= \nabla_y f + \lambda^T \nabla_u d \, D_u^{-1} T \\ &= \nabla_y f \left[D_u T \, D_u^{-1} T \right] + \lambda^T \nabla_u d \, D_u^{-1} T \\ &= \left[\nabla_y f \, D_u T + \lambda^T \nabla_u d \right] D_u^{-1} T \\ &= D_u \mathcal{L}^U \, D_u^{-1} T. \end{aligned}$$

Since $D_u T$ is invertible, a point (u^*, λ^*) is a stationary point of $\mathcal{L}^U$ if and only if $(T(u^*), \lambda^*)$ is a stationary point of $\mathcal{L}^Y$. Similar reasoning applies to the primal and dual feasibility conditions ($d(u^*) \leq 0$ and $\lambda^* \geq 0$) as well as to the complementarity condition ($\lambda_i^* d_i(u^*) = 0$). This completes the first part of the proof.

We now analyze the KKT points. For this, rather than using the definition of invexity directly, we will use properties of the tangent cones. In this way, we will also be able to reason about points on the boundary of the feasible set; recall that invexity is defined on open sets, *i.e.*, sets without their boundary (cf. Definition 5.2).

We first recall a few definitions. Relevant references are Aubin and Ekeland (1984), Geiger and Kanzow (2013), Nocedal and Wright (2006).

Definition 5.5. *The set*

$$\mathcal{A}(y^*) := \{i \in \mathcal{I} : (d_i \circ T^{-1})(y^*) = 0\}$$

is called the active set *for the problem* $(\mathcal{P}^Y)$ *at the point* $y^* \in Y$.

Definition 5.6. *The set*

$$\mathcal{F}(y^*) := \{t \in \mathbb{R}^n : t^T D_y(d_i \circ T^{-1})(y^*) \leq 0 \quad \forall i \in \mathcal{A}(y^*)\},$$

is called the set of linearized feasible directions *for the problem* $(\mathcal{P}^Y)$ *at the point* $y^* \in Y$.

Definition 5.7. *The cone*

$$K^\circ := \{y \in \mathbb{R}^n : y^T x \leq 0 \quad \forall x \in K\}$$

is called the polar cone *of the cone* K.

Let (u^*, λ^*) be a KKT point of $(\mathcal{P}^U)$. Our aim is to show that u^* is global minimum of f on the set $V(u^*)$ as defined in (5.8). For this,

it is convenient to reason about $y^* = T(u^*)$ and problem $(\mathcal{P}^Y)$. By Condition 2 of Definition 5.1, the LICQ holds for the constraint function $d \circ T^{-1}$. Therefore, $\mathcal{F}(y^*) = \mathcal{T}_Y(y^*)$. See (Nocedal and Wright 2006, Theorem 12.1) for proof of this fact.

Since, by the first part of the proof, (y^*, λ^*) is also KKT point, we have

$$-\nabla_y f(y^*) = \lambda^{*T} D_y(d \circ T^{-1})(y^*). \tag{5.9}$$

Following the definition of the set of linearized feasible directions $\mathcal{F}(y^*)$, for all $t \in \mathcal{F}(y^*) = \mathcal{T}_Y(y^*)$ we have that $t^T D_y(d_i \circ T^{-1})(y^*) \leq 0$ for all $i \in \mathcal{A}(y^*)$. Because of this and the facts that $\lambda_i^* \geq 0$ for all $i \in \mathcal{A}(y^*)$ and $\lambda_i^* = 0$ for all $i \in \mathcal{I} \setminus \mathcal{A}(y^*)$, it follows from (5.9) that $-t^T \nabla_y f(y^*) \leq 0$ for all $t \in \mathcal{F}(y^*) = \mathcal{T}_Y(y^*)$. Therefore $-\nabla_y f(y^*) \in (\mathcal{T}_Y(y^*))^\circ$, the polar cone of the tangent cone.

Since $T[V(u^*)] \subset Y$, it follows directly from Definition 5.3 that $\mathcal{T}_{T[V(u^*)]}(y^*) \subset \mathcal{T}_Y(y^*)$. The inclusion reverses when taking polar cones, so that

$$-\nabla_y f(y^*) \in (\mathcal{T}_Y(y^*))^\circ \subset (\mathcal{T}_{T[V(u^*)]}(y^*))^\circ.$$

This means that for every tangent vector $t \in \mathcal{T}_{[V(u^*)]}(y^*)$, we have $t^T \nabla_y f(y^*) \geq 0$. By convexity of f (Condition 5 of Definition 5.1), for every $y \in T[V(u^*)]$,

$$f(y) - f(y^*) \geq (y - y^*)^T \nabla_y f(y^*) \geq 0.$$

The second inequality follows from the fact that $y - y^* \in \mathcal{T}_Y(y^*)$ by construction of the set $V(u^*)$. We conclude that y^* is a global minimum of f on $T[V(u^*)]$, whence u^* is a global minimum of $f \circ T$ on $V(u^*)$. □

5.4 Numerical experiment: river water level control

5.4.1 Introduction

In this section, we describe a numerical experiment involving a discrete-time optimization problem for a system driven by hyperbolic partial differential equations. The experiment illustrates how a search for a KKT point using a local solver always leads to the same conic-intersection optimum. This suggests that this solution is in fact a conic-intersection or global optimum, as predicted by the theory. In turn, this highlights the practical relevance of our result: without further analysis, one would only be able to claim any global properties of a solution using computationally expensive general-purpose solvers for non-convex problems such as, *e.g.*, COUENNE (Belotti et al. 2009), or, alternatively, seeding a local search with a large number of different starting points in order to obtain increased confidence in the quality of the solution.

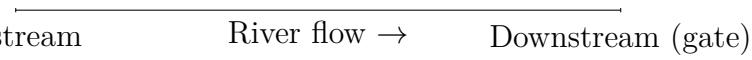


Figure 5.4 – Conceptual view of river segment.

We consider a single river segment under the decision maker's control, illustrated in Figure 5.4. The river segment has two endpoints: (i) the upstream endpoint – flow through this point over time is treated as an external parameter for which a time series is available, (ii) the downstream endpoint – a gate where we control the outflow with decision variables. The goal is to schedule the water release at the downstream endpoint over time so that the level over the river segment deviates as little as possible from the target level. This models the control problem faced by operators of impounded rivers, *i.e.*, rivers where segments are separated by weirs or dams, such as the Meuse and the Moselle in Europe.

Given the upstream inflow, downstream outflow, and initial conditions, the time evolution of the level and flow of water at each point of the river is governed by hyperbolic partial differential equations (Vreugdenhil 2013). These are known as the *Saint-Venant* equations, and are given by the momentum equation

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}\frac{Q^2}{A} + gA\frac{\partial H}{\partial x} + g\frac{Q|Q|}{ARC^2} = 0, \tag{5.10}$$

with the longitudinal coordinate x increasing in the flow direction of the river, time t, flow (discharge) Q, water level H, cross section A, hydraulic radius $R := A/P$, wetted perimeter P, Chézy friction coefficient C, gravitational constant g, and by the mass balance (or continuity) equation

$$\frac{\partial Q}{\partial x} + \frac{\partial A}{\partial t} = 0.$$

In a setting with bidirectional flow, the $|Q|$ factor in the momentum equation (5.10) may be approximated by a smooth function (Burgschweiger et al. 2009). In this section, however, we will only consider unidirectional flow with $Q > 0$, so that $|Q| = Q$. We will also not consider wetting and drying of the channel, whence we assume that $A > 0$ and $P > 0$.

5.4.2 Discretization and analysis

In order to build up a finite-dimensional optimization problem, we conduct a spatial discretization with $N + 1$ equidistant points at which the flows Q_i are computed. Between each pair of adjacent flow computation points we consider the level H_i points, such that H_i is always between Q_{i-1} and Q_i. For each of these points, we consider a time discretization with T equidistant steps. We denote the corresponding flow and level over time by $Q_i(t_j)$ and $H_i(t_j)$. This staggered grid is illustrated in Figure 5.5.

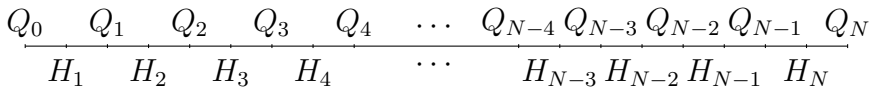


Figure 5.5 – Staggered grid for the example problem.

The approximation scheme is semi-implicit in time, following Casulli and Cheng (1990). Semi-implicit means that for the equations at time step t_j, some terms are evaluated explicitly at time step t_{j-1}, and other terms are evaluated implicitly at time step t_j, requiring the solution of a system of equations to determine their values. The semi-implicit schemes of Casulli and Cheng (1990) and Casulli and Zanolli (1998) strike a balance between numerical stability on the one hand, and non-singular tridiagonal Jacobian structure on the other hand.

The discretized mass balance equation is

$$\frac{Q_i(t_j) - Q_{i-1}(t_j)}{\Delta x} + \frac{A_i(H_i(t_j)) - A_i(H_i(t_{j-1}))}{\Delta t} = 0, \qquad (5.11)$$

and the discretized momentum equation is

$$\frac{Q_i(t_j) - Q_i(t_{j-1})}{\Delta t} + e_{i,j} + gA_{i+\frac{1}{2}}(t_{j-1})\frac{H_{i+1}(t_j) - H_i(t_j)}{\Delta x} + g\frac{P_{i+\frac{1}{2}}(t_{j-1})\, Q_i(t_{j-1})Q_i(t_j)}{A_{i+\frac{1}{2}}(t_{j-1})^2\, C_i^2} = 0, \qquad (5.12)$$

with

$$A_{i+\frac{1}{2}}(t_j) := \frac{1}{2}\left(A_i(H_i(t_j)) + A_{i+1}(H_{i+1}(t_j))\right),$$
$$P_{i+\frac{1}{2}}(t_j) := \frac{1}{2}\left(P_i(H_i(t_j)) + P_{i+1}(H_{i+1}(t_j))\right),$$

spatial index i, temporal index j, spatial step size Δx, temporal step size Δt, and convective acceleration $e_{i,j}$. The parameter C_i indicates the local friction coefficient, and H_i^b indicates the local

bottom level. The convective acceleration term $e_{i,j}$ is discretized explicitly in time as

$$e_{i,j} := \frac{2Q_i(t_{j-1})}{A_{i+\frac{1}{2}}(t_{j-1})} \frac{Q_i(t_{j-1}) - Q_{i-1}(t_{j-1})}{\Delta x} - \frac{Q_i(t_{j-1})^2}{A_{i+\frac{1}{2}}(t_{j-1})^2} \frac{A_{i+1}(H_{i+1}(t_{j-1})) - A_i(H_i(t_{j-1}))}{\Delta x}.$$

Similar to Example 5.2.10, the equations are such that they can be solved forward in time, starting from initial and boundary conditions. Some boundary conditions may be fixed a priori, whereas others may be given by control variables. In the example grid of Figure 5.5, Q_0 is a fixed boundary condition, whereas Q_N is controlled. The values at time step t_0 are taken fixed; these are the initial conditions.

We define the optimization problem as

$$\begin{aligned} \min_{\substack{Q_i(t_j) \\ H_i(t_j)}} \; & f = \sum_{j=1}^{T} |H_N(t_j) - \overline{H}_N|^2 \qquad (\mathcal{RP}) \\ \text{s.t. } & (5.11) - (5.12) \\ & \underline{Q} \leq Q_N(t_j) \leq \overline{Q} \quad \forall j \in \{1, \ldots, T\}, \end{aligned}$$

where the objective is to keep the level at the H node upstream of the gate, *i.e.*, H_N, as close as possible to the target level $\overline{H}_N$. The function of the remaining states $H_i(t_j)$, $i < N$, is to ensure physically accurate wave propagation in the up- and downstream directions.

First, we relate problem $(\mathcal{RP})$ to the general problem $(\mathcal{P})$. The decision variables $Q_N(t_j)$ play the role of the control vector u, the remaining variables $Q_i(t_j)$ where $i < N$ and $H_i(t_j)$ play the role of the state vector x, and the output vector y consists of the subset of state variables $H_N(t_j)$.

We now proceed to analyze the problem ($\mathcal{RP}$) in light of the conditions of Definition 5.1.

Condition 1 follows from the fact that the Jacobian of the equations (5.11) – (5.12) with respect to the states at time step t_j is tridiagonal. In (Casulli and Cheng 1990, p. 128) this Jacobian is shown to be non-singular, thanks to the semi-implicit approximation scheme. The equations (5.11) – (5.12) do not depend on future states at time step t_k, $k > j$. Therefore, the full rank of the Jacobian $\nabla_x c$, *i.e.*, the Jacobian of all equality constraints c with respect to the complete state vector x, follows by induction on the final time index T.

Condition 2 follows from the linearly independent inequality constraints on the outflow variables $Q_N(t_j)$.

Condition 3 follows from the fact that, given states at time step t_{j-1}, the states at time step t_j are uniquely defined. If the cross section A is a linear function of the water level H, this follows readily: in this case the equations (5.11) – (5.12) are linear in the states at time step t_j, and the Jacobian to these states is full-rank as per Condition 1. In Casulli and Zanolli (1998) it is shown that the uniqueness result also extends to a class of nonlinear, monotonic cross section functions $A = A(H)$.

Condition 4 follows from two considerations. First of all, only equation (5.11) depends on the control variables. Since the dependency is linear, and every control variable occurs in exactly one constraint (5.11), the Jacobian $\nabla_u c$ is full-rank. Secondly, for a given control variable $Q_N(t_j)$, only the states at t_k, $k \geq j$ are sensitive to it. Consequently, the mapping from the control variables $Q_N(t_j)$ to the output variables $H_N(t_j)$ is injective.

Condition 5 follows from the fact that the objective function f in problem ($\mathcal{RP}$) is convex in the output variables $H_N(t_j)$.

Table 5.1 – Parameters for the example problem.

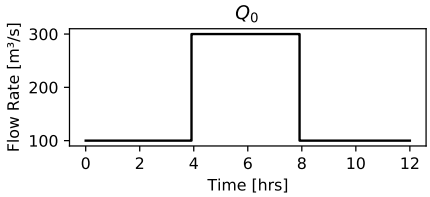
Parameter	Value	Description
T	72	Index of final time step
Δt	600 s	Time step size
H_i^b	$(-4.90, -4.92, \ldots, -5.10)$ m	Bottom level
l	10 000 m	Total channel length
$A_i(H_i)$	$50 \cdot (H - H_i^b)$ m^2	Channel cross section function
$P_i(H_i)$	$50 + 2 \cdot (H - H_i^b)$ m	Channel wetted perimeter function
C_i	$(40, 40, \ldots, 40)$ m$^{0.5}$/s	Chézy friction coefficient
$H_i(t_0)$	$(0.000, -0.025, \ldots, -0.222)$ m	Initial water levels at H nodes
$Q_i(t_0)$	$(100, 100, \ldots, 100)$ m^3/s	Initial flow at Q nodes
$\underline{Q}$	100 m^3/s	Minimum outflow
$\overline{Q}$	200 m^3/s	Maximum outflow
$\overline{H}_N$	0 m	Target water level
N	10	Number of level computation points

5.4.3 Application

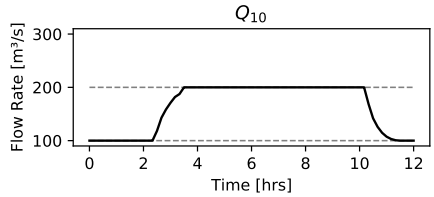
We now present a concrete problem, based upon the experimental setting from the draft Baayen et al. (2020). In this problem, we consider a single river segment with $N = 10$ uniformly spaced level nodes and rectangular cross section, an upstream inflow boundary condition provided with a fixed time series, as well as a controllable downstream release boundary condition. The hydraulic parameters and initial conditions are summarized in Table 5.1. The model starts from steady state: the initial flow rate is uniform and the water level decreases linearly along the length of the channel. Our objective is to keep the water level at the H node upstream of the gate, *i.e.*, H_N, at the target value $\overline{H}_N = 0$ m.

A solution to the optimization problem was obtained using the interior point solver IPOPT (Wächter and Biegler 2006) and is plotted in Figure 5.6. By releasing water in anticipation of the inflow using the decision variable Q_N, the optimization is able to reduce water level fluctuations and keep the water levels close to the target.[1]

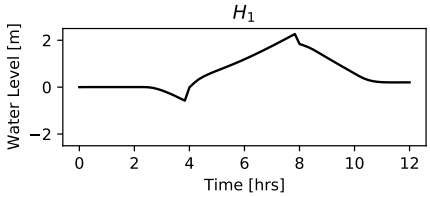
[1] An example implementation in Python using the algorithmic differen-



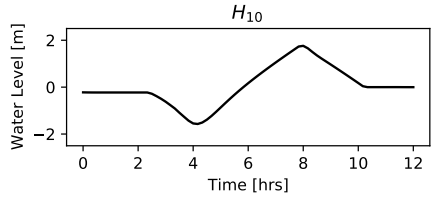
(a) Fixed inflow boundary condition.



(b) Outflow bounds (dashed) and trajectory.



(c) Upstream level trajectory.



(d) Downstream level trajectory.

Figure 5.6 – Solution to the example problem.

The IPOPT solve, starting from an all-zero starting point, takes approximately 0.1 s to complete on a 2.6 GHz Intel Core i7 CPU.

Since some of the bounds on the control variables are active, the conic intersection $V(u^*)$ of Theorem 5.1 need not cover the entire set U. The solution quality was analyzed numerically by seeding the optimization with a large number of different starting points. Latin hypercube sampling (McKay et al. 2000) was used to compute 1 000 different starting points, for each of which IPOPT computed a solution. The standard deviation of the solution vectors was found to be in the order of $10^{-11} - 10^{-13} \approx 0$ per solution vector coordinate, illustrating how every starting point resulted in an – for all practical purposes – identical solution. This provides evidence that the found solution is in fact globally optimal, or at least nearly so.

There is also ample other numerical evidence that solutions of this type are globally optimal or very close to it. In Baayen et al. (2019b), the solution quality and run time of an interior point-type method (IPM) is benchmarked against a so-called *reduced genetic algorithm* (RGA, Vermuyten et al. (2018)) for a class of water optimization problems. The IPM search finds qualitatively consistent solutions that always obtain better objective function values than the RGA. This benchmark includes problems with multiple river segments, multiple spatial control points, and both coarser and finer discretizations of the Saint-Venant equations in time and space.

Similar results are reported in Ghaddar et al. (2017) for drinking water distribution networks, where local search using IPOPT

---

tiation package CasADi (Andersson et al. 2019) and homotopy continuation (Allgower and Georg 2012, Baayen et al. 2019a) to solve the nonlinear equality constraints numerically using IPOPT, is available online at `https://github.com/jbaayen/homotopy-example`.

finds solutions with objective values within a relative distance of 10^{-3} of those found using the global solver COUENNE – in a fraction of the computation time. The general-purpose global solvers require multiple minutes or hours to run, but the local search completes in a few seconds at most.

5.5 Conclusions and outlook

In this work, we used the invariant convexity (invexity) to provide a theoretical framework to understand the good (often optimal) performance of local solvers on a class of non-convex problems that arise in discrete-time optimal control. We showed that KKT points for such problems are *conic-intersection optima*. Conic-intersection optimality implies global optimality for solutions in the interior of the feasible set, and is somewhat weaker – yet significantly stronger than local optimality – on the boundary.

For a concrete PDE-constrained problem that is representative of a real-world water management problem, we demonstrated hidden invexity and verified the high quality of solutions obtained by local search numerically. To the best of our knowledge, this is the first time that the theory of invex functions has been leveraged in a large-scale applied engineering context.

To make our results applicable beyond the domain of discrete-time optimal control (where an expert can typically check whether the conditions of Definition 5.1 hold for a given discretization), there are two natural next steps that, however, fall outside the scope of this work. First, it would be of great help to provide automated tools for detecting hidden invexity. Second, one could also investigate the use of hidden invexity to accelerate branch-and-bound algorithms for general non-convex global optimization.

On an analytical level, another interesting direction would be to

extend our result to continuous-time optimal control.

Last but not least, we note that the problems described in the previous chapters are regular with respect to their continuous decision variables. Integer decision variables can be solved for using a branch-and-bound scheme, as described in the draft Baayen and Marecek (2020). Hidden invexity, therefore, obsoletes the need for the approximations of Chapters 3 and 4. Furthermore, hidden invexity allows water allocation applications, such as the application of Chapter 2, to be built using a physically accurate hydraulic description of the water system (instead of using coarse linear models). Perhaps most excitingly, hidden invexity allows water optimization problems to be based on existing, calibrated hydraulic models set up using industry-standard packages such as SOBEK (Deltares 2016) or HEC-RAS (Brunner 2002).

5.5 Acknowledgements

The authors thank Vyacheslav Kungurtsev for the critical reading of the manuscript and his comments.

CHAPTER 6

Benchmark: Open channel water level control

6.1 Introduction

In this chapter we benchmark the performance of the local solver IPOPT against the performance of RGA (Vermuyten et al. 2018), a type of genetic algorithm, for a suite of open channel optimization problems. Each of the problems in the suite is an extension of Problem ($\mathcal{RP}$) described in Section 5.4. These problems are regular in the sense of Definition 5.1 and as such, we know the IPOPT solutions to be conic-intersection optima as per Theorem 5.1.

In Section 6.2 we describe the benchmark in detail. In Section 6.3 the results are presented and discussed. Finally, in Section 6.4, we draw conclusions.

6.2 Benchmark setup

Every problem in the benchmark suite is a multiple of Problem ($\mathcal{RP}$) in the sense that it contains river reaches with flow dynamics governed by Saint-Venant equations joined together with flow

control variables. In this way, the regularity analysis for $(\mathcal{RP})$ of Section 5.4.2 carries over to the multi-reach problems. Every flow control variable is understood as the simplest possible representation of a weir. This setup is illustrated for a 2-weir probem in Figure 6.1. The discretization grid for the 2-weir problem is illustrated in Figure 6.2, and the upstream inflow boundary condition, Q_0, is shown in Figure 6.3.

Figure 6.1 – Conceptual view of a river with two reaches and two weirs.

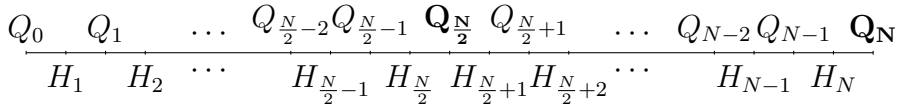


Figure 6.2 – Staggered grid for a 2-weir problem. The control variables are $Q_{\frac{N}{2}}$ and Q_N.

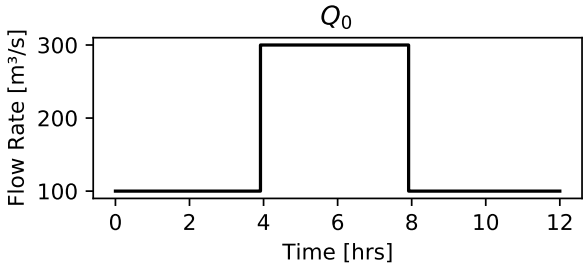


Figure 6.3 – Inflow boundary condition Q_0.

The geometry of each reach is as described in Table 5.1, and the objective is to track a reference water level of 0 m as in Problem $(\mathcal{RP})$. The hydraulic and control time steps are decoupled, such that the control time step size can be a multiple of the hydraulic

Table 6.1 – Parameter variations for the benchmark.

Parameter	Values
Total number of computation points	16, 32, 64, 128, 256, 512
Number of weirs	1, 2, 4, 8, 16
Hydraulic time step	5 min, 10 min, 15 min
Control time step	1 hour, 2 hours, 4 hours, 8 hours

step size. For this benchmark, parameters were varied according to Table 6.1, resulting in a total of 360 configurations.

The total number of computation points per river reach is equal to the total number of computation points, divided by the number of weirs.

As in Section 5.4, IPOPT (Wächter and Biegler 2006) was used as the local solver. RGA was implemented in Fortran following the description of the algorithm in Vermuyten et al. (2018), and the settings of the algorithm were taken from the table of "optimal parameter settings" in said paper. The variability of the RGA results that we observe is in line with the variability reported by the authors.

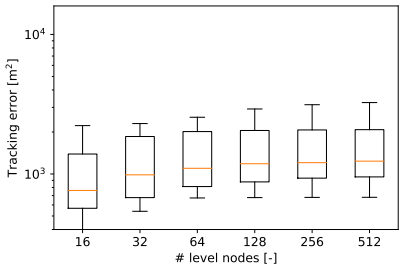
Every configuration was solved 10 times, in order to capture any variance in solution quality and run time. The experiment was carried out in September 2019 on an Amazon Web Services EC2 "c4.xlarge" node with 16 Intel Xeon cores at 3 GHz, 32 GB of RAM, and running Ubuntu Linux 18.04.3 LTS.

6.3 Results

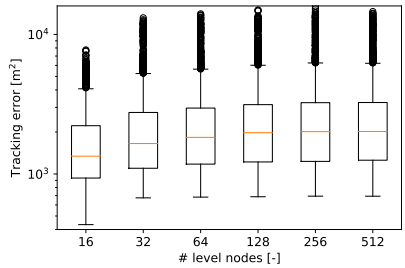
In the following, we present and discuss the results: first, with respect to the tracking error, and secondly with respect to the wall time.

6.3.1 Tracking error

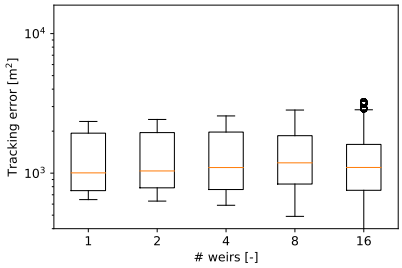
The tracking error measures how close the system is to the target water level, squared and summed over all time steps and over the discretization points upstream of the weirs. It is identical to the objective function value of the benchmark problems (cf. Problem ($\mathcal{RP}$)). Figures 6.4 and 6.5 present the tracking error in relation to the number of computational nodes, number of weirs, hydraulic step size, and control step size, respectively. Because our benchmark data set lives in a 6-dimensional space (4 parameters and 2 indicators), the plots provide projections of the results onto various 2-dimensional planes. Every plot relates a parameter to an indicator.



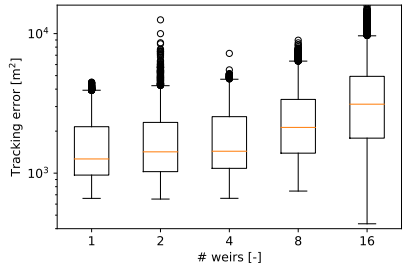
(a) Number of level nodes vs. tracking error (IPOPT).



(b) Number of level nodes vs. tracking error (RGA).

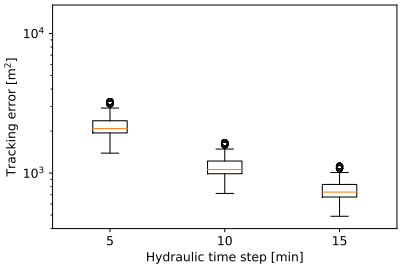


(c) Number of weirs vs. tracking error (IPOPT).

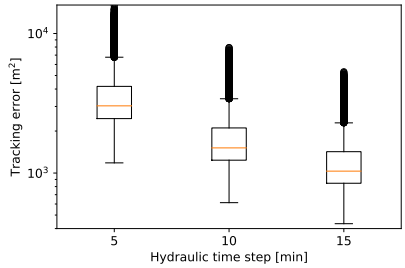


(d) Number of weirs vs. tracking error (RGA).

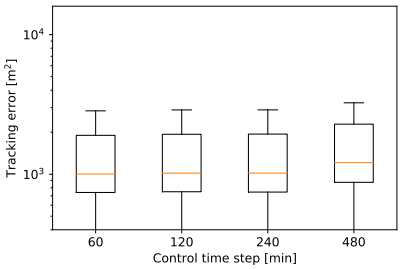
Figure 6.4 – Comparison of benchmark results: tracking error (I).



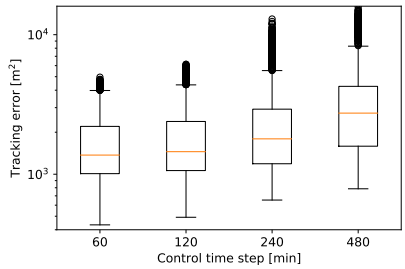
(a) Hydraulic time step vs. tracking error (IPOPT).



(b) Hydraulic time step vs. tracking error (RGA).



(c) Control time step vs. tracking error (IPOPT).



(d) Control time step vs. tracking error (RGA).

Figure 6.5 – Comparison of benchmark results: tracking error (II).

We may observe how the tracking error of the RGA results exhibits substantial variance. This is expected, given the randomized nature of the RGA algorithm. In contrast, the tracking error of the IPOPT result shows little variance. Furthermore, the tracking error of the RGA results worsens as the number of weirs (control variable) is increased (Figure 6.4d). This effect is largely absent for the IPOPT results (Figure 6.4c). A third effect is how the tracking error of the RGA results worsens as the control step size is increased (Figure 6.5d). This effect is also largely absent for the IPOPT results (Figure 6.5c).

Figures 6.6–6.10 report the tracking errors for each individual configuration, grouped by the number of weirs. In these figures, we see clearly how the IPOPT solutions do not, in fact, exhibit any variance at all. We also see how the tracking error, *i.e.*, the objective function value, is always superior to the results obtained using RGA. This provides a clear indication that the solutions found by IPOPT are globally optimal, or nearly so.

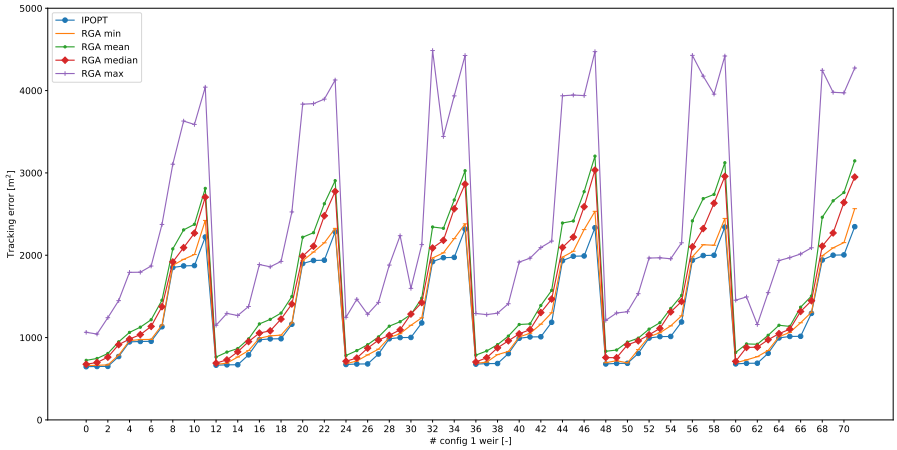


Figure 6.6 – Tracking error for all parameter configurations with 1 weir.

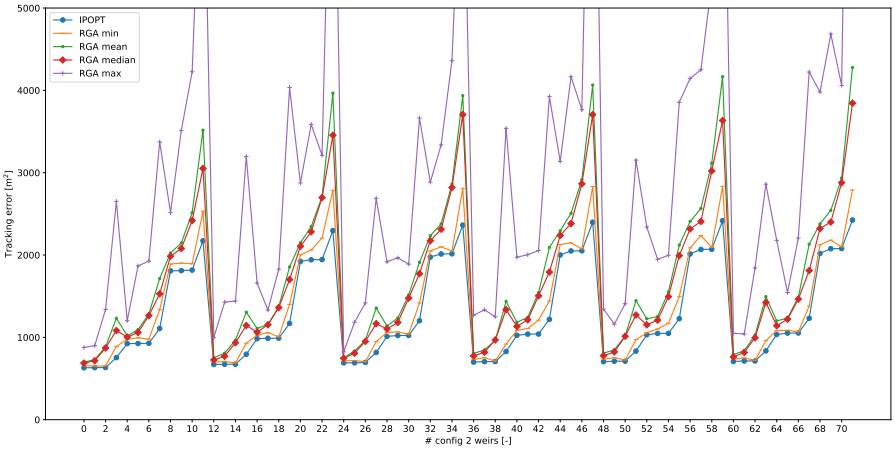


Figure 6.7 – Tracking error for all parameter configurations with 2 weirs.

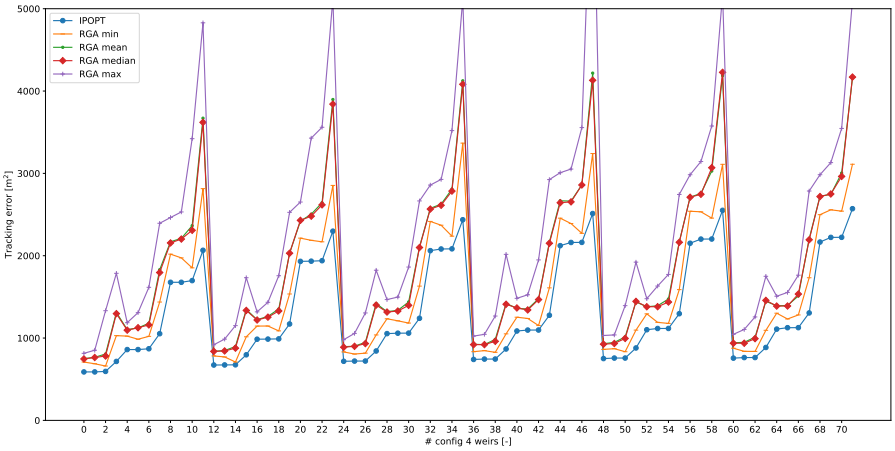


Figure 6.8 – Tracking error for all parameter configurations with 4 weirs.

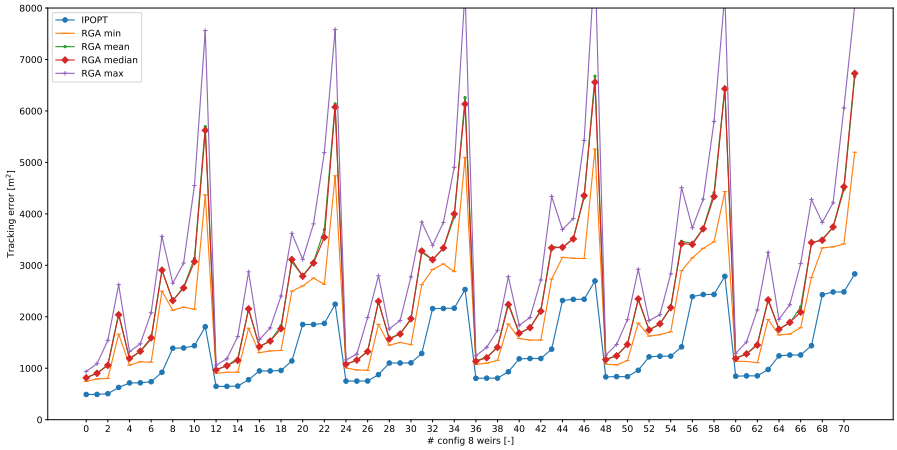


Figure 6.9 – Tracking error for all parameter configurations with 8 weirs.

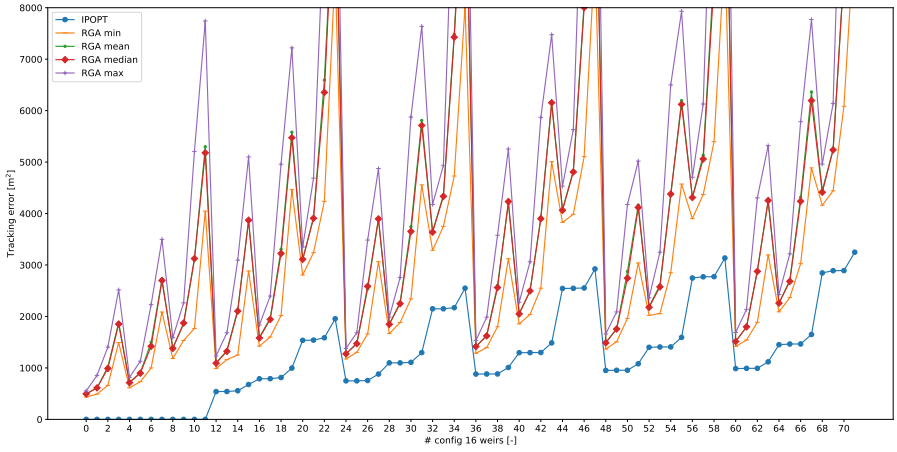
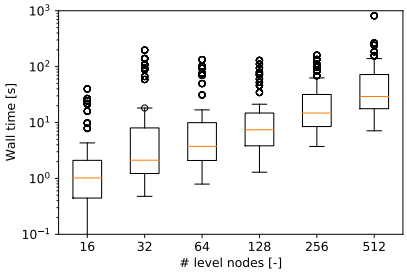


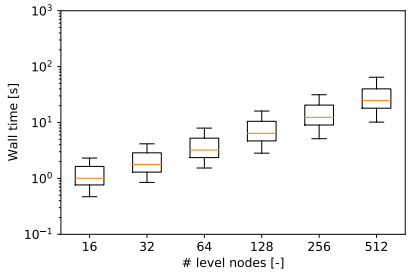
Figure 6.10 – Tracking error for all parameter configurations with 16 weirs.

6.3.2 Run time

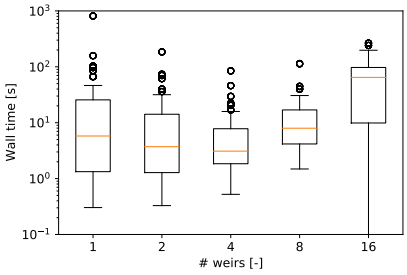
The wall time is the difference between the system time when the optimization run completed, and the system time when the run started. It is a more intuitive measure of run time than CPU or process time. Figures 6.11 and 6.12 present the wall time in dependence of the number of computational nodes, number of weirs, hydraulic step size, and control step size.



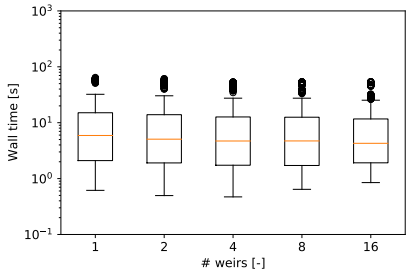
(a) Number of level nodes vs. wall time (IPOPT).



(b) Number of level nodes vs. wall time (RGA).

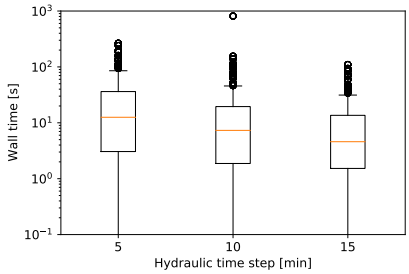


(c) Number of weirs vs. wall time (IPOPT).

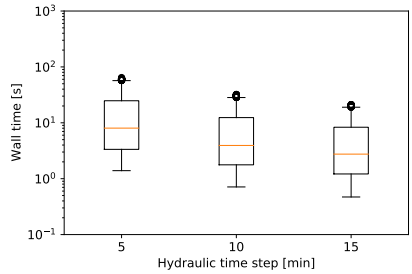


(d) Number of weirs vs. wall time (RGA).

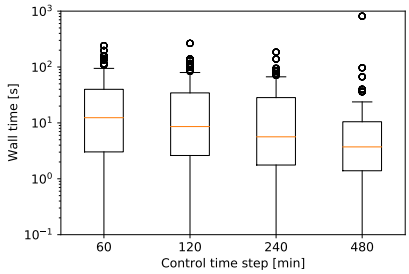
Figure 6.11 – Comparison of benchmark results: wall time (I).



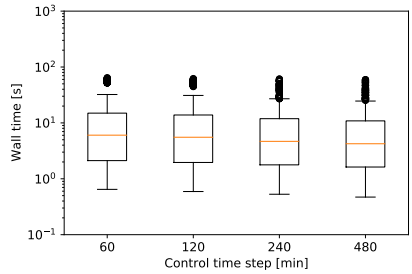
(a) Hydraulic time step vs. wall time (IPOPT).



(b) Hydraulic time step vs. wall time (RGA).



(c) Control time step vs. wall time (IPOPT).



(d) Control time step vs. wall time (RGA).

Figure 6.12 – Comparison of benchmark results: wall time (II).

Here, we observe an advantage of the RGA algorithm. RGA runs sometimes (but not always) finish in less time than IPOPT. The RGA run time also exhibits less variance. In general, however, the median run times of both algorithms are in the same order of magnitude.

6.4 Conclusions

Our benchmark demonstrates how IPOPT applied to the open channel water level control problem yields solutions with excellent quality. The solutions appear to be globally optimal, or nearly so.

This is in line with the regularity of these problems and the hidden invexity results of Chapter 5, thereby providing further numerical evidence for the theoretical results.

A drawback of IPOPT is the variance in wall time. Future research may address the development of a local solver with the ability to terminate with a usable solution within a fixed amount of time.

CHAPTER 7

Case study: Hoogheemraadschap van Rijnland

In this chapter a *real-life* application is discussed, in which the high quality of the solutions is supported by the hidden invexity results of Chapter 5. This application is a decision support system (DSS) developed for the district water authority *Hoogheemraadschap van Rijnland*, in the west of the Netherlands.

In the Netherlands, *Rijkswaterstaat*, the Dutch national authority responsible for the national waterways and water bodies, and the district water authorities are charged with general water management. Among other tasks, they are responsible for ensuring a sufficient supply of water and keeping the country protected against flooding. The area under the authority of Rijnland covers the area between Amsterdam in the North and The Hague in the South, including Amsterdam Schiphol Airport in the *Haarlemmermeer* polder. See Figure 7.1.

7.1 System description

Water managers at Rijnland run an operational DSS for the *boezem* section of their network (primary canal, main water distribution

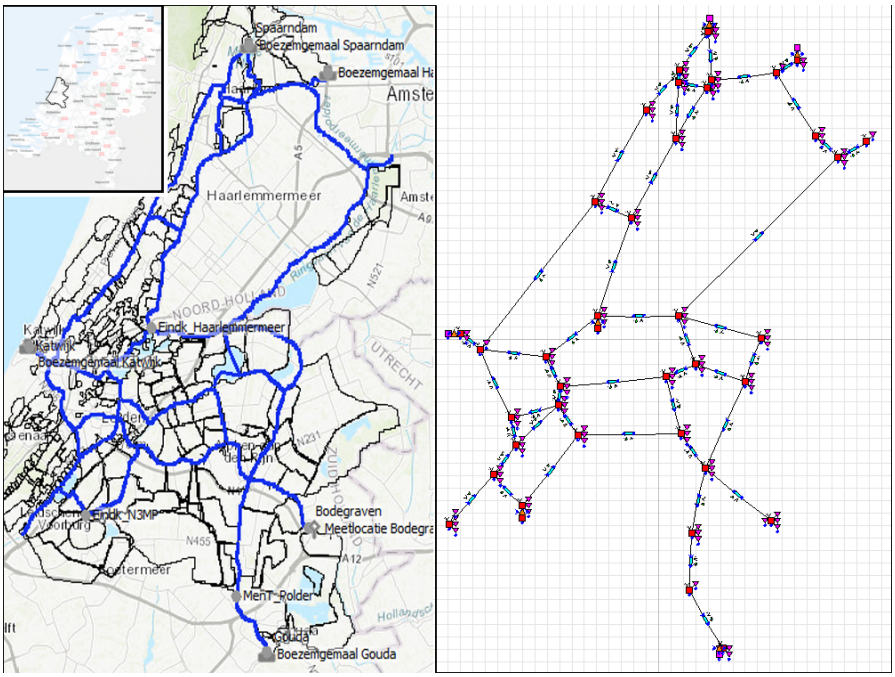


Figure 7.1 – The primary canal of Rijnland, the *boezem*, highlighted in blue (left). The model schematization of the primary canal, with pink squares indicating storage and blue bars representing channel reaches (right).

system). This DSS aims at optimizing the use of the pumping stations with respect to energy, costs and availability, while making sure that various operational goals are met. The responsibilities of Rijnland include the following: flood defense and water quantity, water quality, waste water and ground water. Which operational goals should be met depends on the season, meteorological conditions and current state of the system. Besides the polder pumps, the water managers have four large pumping stations at their disposal, with a total discharge capacity of 199 m^3/s (17.2×10^6 m^3/day or 15.9 mm/day).

In wet periods, the water is mainly pumped from polders (the low lying areas) to the boezem. Surplus water is pumped towards the outside water system. The main objectives of the optimization model relate to maintaining water levels between acceptable bounds for a forecast horizon of 48 hours, taking into account the sloping effect of wind on the canal water levels, as well as forecast polder inflow. Secondary goals include minimizing energy use and pump cost, by using the available buffers in the canal system to schedule the absolute pump discharges with respect to energy prices and outside water levels.

In dry periods, the system calculates the effect of water distribution towards the polders, originating from a branch of the Rhine river. Like during the wet season, the main objectives of the optimization model relate to maintaining water levels between acceptable bounds for a forecast horizon of 48 hours, taking into account the wind effects and forecast polder outflow. An important secondary goal is taking the salinity of the water and mass transport into account. It is vital that the salinity levels at certain key points in the system stay within bounds, to ensure no damage to the intensive agricultural and horticulture practices. When needed, the system should optimize for the quantity and time of fresh water intake (with a low chloride concentration) from the Hollandse IJssel river at pumping station Gouda.

Some key characteristics for the Rijnland system are:

- $1,175$ km^2,
- $1,300,000$ inhabitants,
- $523,500$ households,
- 30 municipalities over 2 provinces,

- 90% below sea level,
- Annual precipitation: 850 mm,
- Annual evaporation: 550 mm,
- 202 polders (5.5% open water).

7.2 Model setup

The Rijnland boezem system is modeled in the Modelica language (Elmqvist 1997) and the RTC-Tools Channel Flow model component library (den Toom et al. 2018).

In order to be able to take the sloping effects of the wind into account, the system was modelled hydraulically. The inertial wave equation (1.4) was chosen for numerical simplicity, and was determined experimentally to be sufficiently accurate for this system. Figure 7.2 shows a comparison between solutions to the inertial wave and the full Saint-Venant equations. The model used in this comparison is characteristic of the reaches found in the Rijnland boezem system.

To optimize the control of the Rijnland boezem system, lexicographic goal programming is used to specify a hierarchy of optimization requirements. Higher-priority objectives, such as water level range goals, typically admit multiple equivalent solutions. Lower-priority objectives are optimized within the set of equivalent solutions to the problem corresponding to the previous priority level. The complete goal programming algorithm is explained in Chapter 2.

The goals used in the Rijnland boezem system optimization problem are listed in Table 7.1. Smaller priority numbers take precedence over larger priority numbers. The priority numbers do not

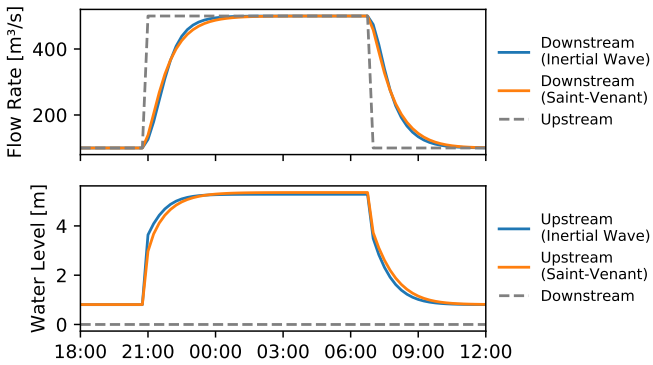


Figure 7.2 – Comparison of solutions to the inertial wave equation (1.4) and for the Saint-Venant momentum equation (1.2). Between 21:00 and 07:00, a flat wave enters the system. The response of the channel reach is shown with flow rates in the top plot, and water levels in the bottom plot.

need to be consecutive. Non-consecutive priority numbers provide the operators with some flexibility to insert goals "in the middle", without needing to renumber the priorities for every lower priority goal.

Example results of a 48h operational optimization run for the Rijnland boezem system are shown in Figure 7.3.

7.3 Results

A DSS using the inertial wave equation implemented with RTC-Tools 2 (Baayen et al. 2018) and IPOPT (Wächter and Biegler 2006) is operational and used daily at Rijnland. It controls the system autonomously in a closed-loop fashion for extended periods of time, without operator intervention. A screenshot of the

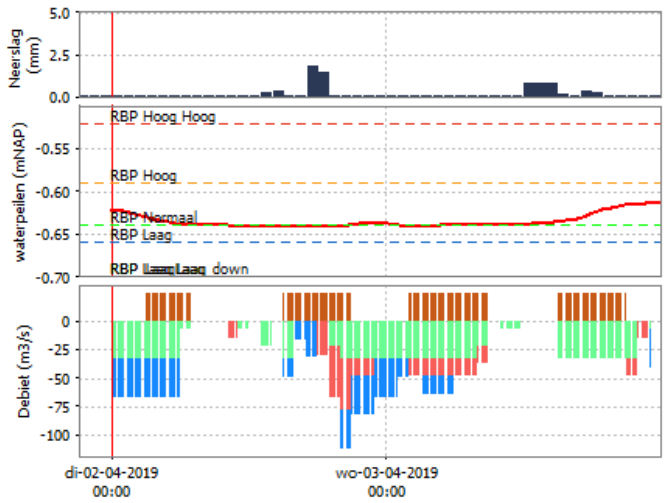


Figure 7.3 – Rainfall forcing (*neerslag*, top). Weighted average simulated water level and alarm levels (*waterpeilen*, middle). Optimized pump discharges, one color per pumping station (*debiet*, bottom).

system, showing water levels in the system and advice on how to operate the pumps of the primary canal, is shown in Figure 7.4.

Further information on this DSS, from the perspective of the Rijnland water authority, can be found in van der Zwan (2017).

The Rijnland DSS bears witness to to the practical relevance of the concept of hidden invexity.

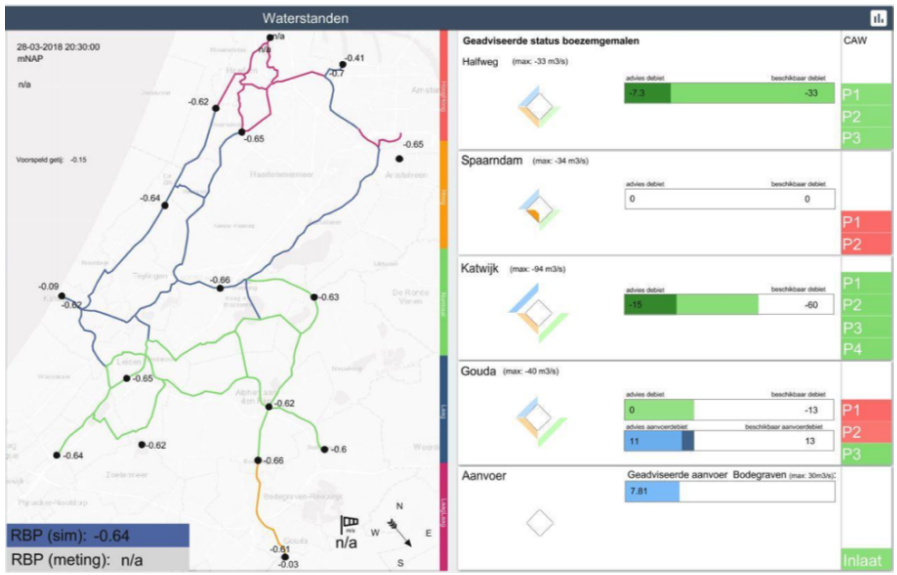


Figure 7.4 – Screenshot of Rijnland DSS. Current water levels are shown color coded on the left, and advice on pump operation is shown on the right.

Table 7.1 – Goals and lexicographical priorities for the Rijnland primary canal optimization problem (smaller priorities take higher precedence). The variable n_p represents the number of pumps, and the variable n_{ep} represents the number of emergency pumps.

Priority	Goal	Description
1	$H_i^{min}(t_j) \leq H_i(t_j) \leq H_i^{max}(t_j)$	Water level range at ten selected nodes
2	$H_{sys}^{min}(t_j) \leq H_{sys}(t_j) \leq H_{sys}^{max}(t_j)$	Water level goal for a weighted average of certain selected nodes
2	$H_{sys}(t_T) = H_{sys}^{setpoint}(t_T)$	Water level setpoint at final timestep for system water level
20	$Q_i^{min}(t_j) \leq Q_i(t_j)$	Saline water flushing related discharge request for pumping station Gouda
30	$\min \sum_{p=1}^{n_{ep}} \sum_{j=1}^{T} Q_{p,j}^2$	Minimization of emergency pump discharge
40	$Q_p(t_j) = Q_p^{setpoint}(t_j)$	General discharge request goals for pumping stations
41	$Q_{katwijk}(t_v) = 0$	Sea tide-related goals for pumping station Katwijk: Discharge request of zero at rising tide (time stamps t_v)
41	$\frac{\partial}{\partial t} Q_{katwijk}(t_w) = 0$	Constant discharge request when sea level above threshold of 0.2m from lowest tide (time stamps t_w)
50	$\min \sum_{p=1}^{n_p} \sum_{j=1}^{T} c_p(t_j) Q_p(t_j)$	Minimization of pumping cost with pump-specific cost time series c_p

Bibliography

T. Ackermann, D. P. Loucks, D. Schwanenberg, and M. Detering. Real-time modeling for navigation and hydropower in the river Mosel. *Journal of Water Resources Planning and Management*, 126(5): 298–303, 2000.

O. Alizadeh-Mousavi and M. Nick. Stochastic security constrained unit commitment with variable-speed pumped-storage hydropower plants. In *Power Systems Computation Conference (PSCC), 2016*, 2016.

E. L. Allgower and K. Georg. *Numerical continuation methods: an introduction*, volume 13. Springer Science & Business Media, 2012.

K.-U. Amann, E. Arnold, and O. Sawodny. Online real-time scheduled model predictive feedforward control for impounded river reaches applied to the Moselle river. In *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 1276–1281. IEEE, 2016.

J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl. CasADi: a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1): 1–36, 2019.

J.-P. Aubin and I. Ekeland. *Applied nonlinear analysis*. John Wiley & Sons, 1984.

J. H. Baayen and J. Marecek. Mixed-integer path-stable optimisation, with applications in model-predictive control of water systems. *arXiv preprint 2001.08121*, 2020.

J. H. Baayen, M. den Toom, O. van Duin, T. Vreeken, J. VanderWees, and D. Schwanenberg. RTC-Tools 2.0, 2018. URL `https://gitlab.com/deltares/rtc-tools`.

J. H. Baayen, B. Becker, K.-J. van Heeringen, I. Miltenburg, T. Piovesan, J. Rauw, M. den Toom, and J. VanderWees. An overview of continuation methods for non-linear model predictive control of water systems. *IFAC-PapersOnLine*, 52(23):73–80, 2019a.

J. H. Baayen, D. J. Vreeken, and P. Archambeau. Optimization methods for hydraulic systems. In *6th international symposium on the hydrological modelling of the Meuse basin*, 2019b.

J. H. Baayen, T. Piovesan, and J. VanderWees. Continuation method for PDE-constrained global optimization: Analysis and application to the shallow water equations. *arXiv:1801.06507*, 2020.

S. A. Bagloee, M. Asadi, and M. Patriksson. Minimization of water pumps' electricity usage: A hybrid approach of regression models with optimization. *Expert Systems with Applications*, 107:222–242, 2018.

P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wächter. Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods & Software*, 24(4-5):597–634, 2009.

A. Ben-Israel and B. Mond. What is invexity? *The ANZIAM Journal*, 28(1):1–9, 1986.

A. Ben-Tal and M. Teboulle. Hidden convexity in some nonconvex quadratically constrained quadratic programming. *Mathematical Programming*, 72(1):51–63, 1996.

P. Bonami, L. T. Biegler, A. R. Conn, G. Cornuéjols, I. E. Grossmann, C. D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. Wächter. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization*, 5(2):186 – 204, 2008. ISSN 1572-5286. In Memory of George B. Dantzig.

S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

J. W. Bronkhorst. De potentie van de meet- en regeltechniek op de rivier de Linge. Bachelor thesis, Delft University of Technology, June 2010.

G. W. Brunner. HEC-RAS (river analysis system). In *North American Water and Environment Congress & Destructive Water*, pages 3782–3787. ASCE, 2002.

J. Burgschweiger, B. Gnädig, and M. C. Steinbach. Optimization models for operative planning in drinking water networks. *Optimization and Engineering*, 10(1):43–73, 2009.

A. Candelieri, R. Perego, and F. Archetti. Bayesian optimization of pump operations in water distribution systems. *Journal of Global Optimization*, 71(1):213–235, 2018.

V. Casulli and R. T. Cheng. Stability analysis of Eulerian-Lagrangian methods for the one-dimensional shallow-water equations. *Applied Mathematical Modelling*, 14(3):122–131, 1990.

V. Casulli and R. T. Cheng. Semi-implicit finite difference methods for three-dimensional shallow water flow. *International Journal for Numerical Methods in Fluids*, 15(6):629–648, 1992. ISSN 02712091.

V. Casulli and P. Zanolli. A conservative semi-implicit scheme for open channel flows. *Journal of Applied Science and Computations*, 5: 1–10, 1998.

G. Cembrano, G. Wells, J. Quevedo, R. Pérez, and R. Argelaguet. Optimal control of a water distribution network in a supervisory control system. *Control engineering practice*, 8(10):1177–1188, 2000.

H. Chanson. *Hydraulics of open channel flow*. Elsevier, 2004.

Y. Collette and P. Siarry. *Multiobjective optimization: principles and case studies*. Springer, 2003. ISBN 3540401822. doi: 10.1007/978-3-662-08883-8.

B. D. Craven. Duality for generalized convex fractional programs. *Generalized concavity in optimization and economics*, pages 473–489, 1981a.

B. D. Craven. Invex functions and constrained local minima. *Bulletin of the Australian Mathematical society*, 24(3):357–366, 1981b.

J. Cunge. On the subject of a flood propagation computation method (musklngum method). *Journal of Hydraulic Research*, 7(2):205–230, 1969.

W. J. De Lange, G. F. Prinsen, J. C. Hoogewoud, A. A. Veldhuizen, J. Verkaik, G. H. O. Essink, P. E. Van Walsum, J. R. Delsman, J. C. Hunink, H. T. L. Massop, et al. An operational, multi-scale, multi-model system for consensus-based, integrated water management and policy analysis: The Netherlands Hydrological Instrument. *Environmental Modelling & Software*, 59:98–108, 2014.

B. Dekens, A. D. Sadowska, P.-J. van Overloop, D. Schwanenberg, and B. De Schutter. Gradient-based hybrid model predictive control using time instant optimization for Dutch regional water systems. In *Control Conference (ECC), 2014 European*, pages 1343–1348. IEEE, 2014.

Deltares. *D-Flow 1D (Sobek 3) Hydrodynamics, Technical reference manual*. Deltares, Delft, The Netherlands, 3 edition, 12 2016.

M. den Toom, J. H. Baayen, and J. VanderWees. RTC-Tools Channel Flow, 2018. URL `https://gitlab.com/deltares/rtc-tools-channel-flow`.

G. F. Dorini, F. Thordarson, P. Bauer-Gottwein, H. Madsen, and D. Rosbjerg. A convex programming framework for optimal and bounded suboptimal well field management. *Water Resources Research*, 48(6), 2012.

H. Elmqvist. Modelica – A unified object-oriented language for physical systems modeling. *Simulation Practice and Theory*, 5(6):p32, 1997. ISSN 09284869. doi: 10.1016/S0928-4869(97)84257-7.

E. A. Eschenbach, T. Magee, E. Zagona, M. Goranflo, and R. Shane. Goal Programming Decision Support System for Multiobjective Operation of Reservoir Systems. *Journal of Water Resources Planning and Management*, 127(2):108–120, 2001.

A. K. V. Falk, C. Mackay, H. Madsen, and P. N. Godiksen. Model

predictive control of a large-scale river network. *Procedia Engineering*, 154:80–87, 2016.

S. A. Fatemi, A. Kuh, and V. Gupta. Energy efficient scheduling algorithms for pumping water in radial networks. In *Information Theory and Applications Workshop (ITA), 2016*, 2017.

O. Fecarotta, A. Carravetta, H. M. Ramos, and R. Martino. An improved affinity model to enhance variable operating strategy for pumps used as turbines. *Journal of Hydraulic Research*, 54(3): 332–341, 2016.

C. A. Floudas. *Nonlinear and mixed-integer optimization: fundamentals and applications*. Oxford University Press, 1995.

C. E. García, D. M. Prett, and M. Morari. Model predictive control: Theory and practice – A survey. *Automatica*, 25(3):335–348, 1989.

C. Geiger and C. Kanzow. *Theorie und Numerik restringierter Optimierungsaufgaben*. Springer-Verlag, 2013.

B. Ghaddar, M. Claeys, M. Mevissen, and B. J. Eck. Polynomial optimization for water networks: Global solutions for the valve setting problem. *European Journal of Operational Research*, 261 (2):450–459, 2017.

M. Haasnoot, H. Middelkoop, A. Offermans, E. Van Beek, and W. P. Van Deursen. Exploring pathways for sustainable water management in river deltas in a changing environment. *Climatic Change*, 115(3):795–819, 2012.

M. Haasnoot, W. Van Deursen, J. H. Guillaume, J. H. Kwakkel, E. van Beek, and H. Middelkoop. Fit for purpose? building and evaluating a fast, integrated model for exploring water policy pathways. *Environmental modelling & software*, 60:99–120, 2014.

M. A. Hanson. On sufficiency of the Kuhn-Tucker conditions. *Journal of Mathematical Analysis and Applications*, 80(2):545–550, 1981.

M. A. Hanson. Invexity and the Kuhn–Tucker theorem. *Journal of Mathematical Analysis and Applications*, 236(2):594–604, 1999.

F. M. Hante, G. Leugering, A. Martin, L. Schewe, and M. Schmidt.

Challenges in optimal control problems for gas and fluid flow in networks of pipes and canals: From modeling to industrial applications. In *Industrial mathematics and complex systems*, pages 77–122. Springer, 2017.

S. S. Hashemi, M. Tabesh, and B. Ataeekia. Ant-colony optimization of pumping schedule to minimize the energy cost using variable-speed pumps in water distribution networks. *Urban Water Journal*, 11(5):335–347, 2014.

K. Horváth, M. Gómez, and J. Rodellar. The effect of the choice of the control variables of the water level control of open channels. In *The 10th IEEE International Conference on Networking, Sensing and Control*, Paris, France, April 10-12. 2013.

K. Horváth, B. P. van Esch, D. J. Vreeken, I. Pothof, and J. H. Baayen. Convex modeling of pumps in order to optimize their energy use. *Water Resources Research*, 55(3):2432–2445, 2019.

B. v. d. Hurk, P. Siegmund, and A. Klein Tank. KNMI'14: climate change scenarios for the 21st century–a Netherlands perspective. *KNMI scientific report (WR 2014-01)*, 2014.

B. Joseph-Duran, M. Jung, C. Ocampo-Martinez, S. Sager, and G. Cambrano. Minimization of Sewage Network Overflow. *Water Resources Management*, 28(1):41–63, 2014.

K. E. Lansey and K. Awumah. Optimal pump operations considering pump switches. *Journal of Water Resources Planning and Management*, 120(1):17–35, 1994.

J. B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on optimization*, 11(3):796–817, 2001.

D. Lee, H. D. Nguyen, K. Dvijotham, and K. Turitsyn. Convex restriction of power flow feasibility sets. *IEEE Transactions on Control of Network Systems*, 6(3):1235–1245, 2019.

X. Litrico and V. Fromion. Frequency modeling of open-channel flow. *Journal of hydraulic engineering*, 130(8):806–815, 2004.

X. Litrico and V. Fromion. *Modeling and control of hydrosystems*. Springer Science & Business Media, 2009.

R. Madani, S. Sojoudi, and J. Lavaei. Convex relaxation for optimal power flow problem: Mesh networks. *IEEE Transactions on Power Systems*, 30(1):199–211, 2014.

H. Mala-Jetmarova, N. Sultanova, and D. Savic. Lost in optimisation of water distribution systems? A literature review of system operation. *Environmental Modelling & Software*, 93:209–254, 2017.

P.-O. Malaterre and J. P. Baume. Optimum choice of control action variables and linked algorithms: Comparison of different alternatives. In *Proceedings of USCID Workshop on Modernization of irrigation water delivery systems*, pages 387–405, Phoenix, Arizona, USA, 1999.

F. Malrait, A. K. Jebai, and K. Ejjabraoui. Power conversion optimization for hydraulic systems controlled by variable speed drives. *Journal of Process Control*, pages 133–146, 2017.

A. Marchi, A. R. Simpson, and M. F. Lambert. Optimization of pump operation using rule-based controls in EPANET2: new ETTAR toolkit and correction of energy computation. *Journal of Water Resources Planning and Management*, 142(7), 2016.

A. Marchi, A. R. Simpson, and M. F. Lambert. Pump operation optimization using rule-based controls. *Procedia Engineering*, 186: 210–217, 2017.

D. Martin. The essence of invexity. *Journal of Optimization Theory and Applications*, 47(1):65–76, 1985.

G. T. McCarthy. The unit hydrograph and flood routing. In *proceedings of Conference of North Atlantic Division, US Army Corps of Engineers, 1938*, pages 608–609, 1938.

M. J. McGregor and J. B. Dent. An application of lexicographic goal programming to resolve the allocation of water from the Rakaia River (New Zealand). *Agricultural Systems*, 41(3):349–367, 1993. ISSN 0308521X. doi: 10.1016/0308-521X(93)90009-Q.

M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the

analysis of output from a computer code. *Technometrics*, 42(1): 55–61, 2000.

R. Menke, E. Abraham, P. Parpas, and I. Stoianov. Approximation of system components for pump scheduling optimisation. *Procedia Engineering*, 119:1059–1068, 2015.

R. Menke, E. Abraham, and I. Stoianov. Modeling variable speed pumps for optimal pump scheduling. In *World Environmental and Water Resources Congress 2016*, pages 199–209, 2016.

R. Misener and C. A. Floudas. Antigone: algorithms for continuous/integer global optimization of nonlinear equations. *Journal of Global Optimization*, 59(2-3):503–526, 2014.

S. K. Mishra and G. Giorgi. *Invexity and optimization*, volume 88. Springer Science & Business Media, 2008.

R. A. Montero, D. Schwanenberg, M. Hatz, and M. Brinkmann. Simplified hydraulic modelling in model predictive control of flood mitigation measures along rivers. *Journal of Applied Water Engineering and Research*, 1(1):17–27, 2013.

R. E. Moore and F. Bierbaum. *Methods and applications of interval analysis*, volume 2. SIAM, 1979.

J. Nicklow, P. Reed, D. Savic, T. Dessalegne, L. Harrell, A. Chan-Hilton, M. Karamouz, B. Minsker, A. Ostfeld, A. Singh, et al. State of the art for genetic algorithms and beyond in water resources planning and management. *Journal of Water Resources Planning and Management*, 136(4):412–432, 2010.

J. Nocedal and S. Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

K. Ogata and Y. Yang. *Modern control engineering*. Prentice-Hall, 1997.

K. Oikonomou, M. Parvania, and S. Burian. Integrating water distribution energy flexibility in power systems operation. In *Power & Energy Society General Meeting, 2017 IEEE*, volume 2018-January, pages 1–5, 2018.

P. Olszewski. Genetic optimization and experimental verification of complex parallel pumping station with centrifugal pumps. *Applied Energy*, 178:527–539, 2016.

L. E. Ormsbee and K. E. Lansey. Optimal control of water supply pumping systems. *Journal of Water Resources Planning and Management*, 120(2):237–252, 1994.

E. Price and A. Ostfeld. Discrete pump scheduling and leakage control using linear programming for optimal operation of water distribution systems. *Journal of Hydraulic Engineering*, 140(6):04014017, 2014.

J. Rodellar, M. Gómez, and J. P. Martín Vide. Stable predictive control of open-channel flow. *Journal of Irrigation and Drainage Engineering*, 115(4):701–713, 1989.

N. V. Sahinidis. BARON: A general purpose global optimization software package. *Journal of Global Optimization*, 8(2):201–205, 1996.

J. Schmidt, W. Kemmetmöller, and A. Kugi. Modeling and static optimization of a variable speed pumped storage power plant. *Renewable Energy*, 111:38–51, 2017.

D. Schwanenberg, B. P. J. Becker, and M. Xu. The open real-time control RTC-Tools software framework for modeling RTC in water resources systems. *Journal of Hydroinformatics*, 17(1):130, 2015. ISSN 1464-7141. doi: 10.2166/hydro.2014.046.

L. Sela Perelman and S. Amin. Control of tree water networks: A geometric programming approach. *Water Resources Research*, 51 (10):8409–8430, 2015.

A. Selvi, A. Ben-Tal, R. Brekelmans, and D. den Hertog. Convex maximization via adjustable robust optimization. *Optimization Online*, 2020.

C. Sepúlveda. *Instrumentation, model identification and control of an experimental irrigation canal*. PhD thesis, Technical University of Catalonia, Barcelona, Spain, 2008.

V. Te Chow. *Open-channel hydraulics*. McGraw-Hill civil engineering series. McGraw-Hill, 1959.

E. Todini. A mass conservative and water storage consistent variable parameter Muskingum-Cunge approach. In *Hydrology and Earth System Sciences Discussions*. European Geosciences Union, 2007.

B. Ulanicki and P. R. Kennedy. An optimization technique for water network operations and design. In *Decision and Control, 1994., Proceedings of the 33rd IEEE Conference on*, volume 4, pages 4114–4115. IEEE, 1994.

R. van der Zwan. Optimizing water quantity and quality with RTC-Tools 2. In *Delft-FEWS International User Days*, 2017.

P.-J. van Overloop. Drainage control in water management of polders in the Netherlands. *Irrigation and Drainage Systems*, 20(1):99–109, 2006a.

P.-J. van Overloop. *Model predictive control on open water systems*. PhD thesis, Delft University of Technology, Delft, The Netherlands, 2006b.

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

E. Vermuyten, P. Meert, V. Wolfs, and P. Willems. Combining model predictive control with a reduced genetic algorithm for real-time flood control. *Journal of Water Resources Planning and Management*, 144(2):04017083, 2018.

C. B. Vreugdenhil. *Numerical methods for shallow-water flow*, volume 13. Springer Science & Business Media, 2013.

A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, Mar 2006. ISSN 1436-4646.

B. T. Wahlin. Performance of model predictive control on ASCE test

canal 1. *Journal of Irrigation and Drainage Engineering*, 130(3): 227–238, 2004.

A. Weerts, G. Prinsen, S. Patzke, W. van Verseveld, H. Berger, and T. Kroon. Operational Water Resources Forecasting System for The Netherlands. In *AGU Fall Meeting Abstracts*, volume 1, page 1498, 2011.

C. Wegley, M. Eusuff, and K. Lansey. Determining pump operations using particle swarm optimization. In *Joint Conference on Water Resource Engineering and Water Resources Planning and Management 2000*, pages 1–6. 2000.

E. Weyer. Multivariable LQ control of an irrigation channel: Experimental results and robustness analysis. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 6642–6647, Dec 2006.

T. Wright and P. Gerhart. *Fluid Machinery: Application, Selection, and Design, Second Edition.* Taylor & Francis, 2009. ISBN 9781420082944.