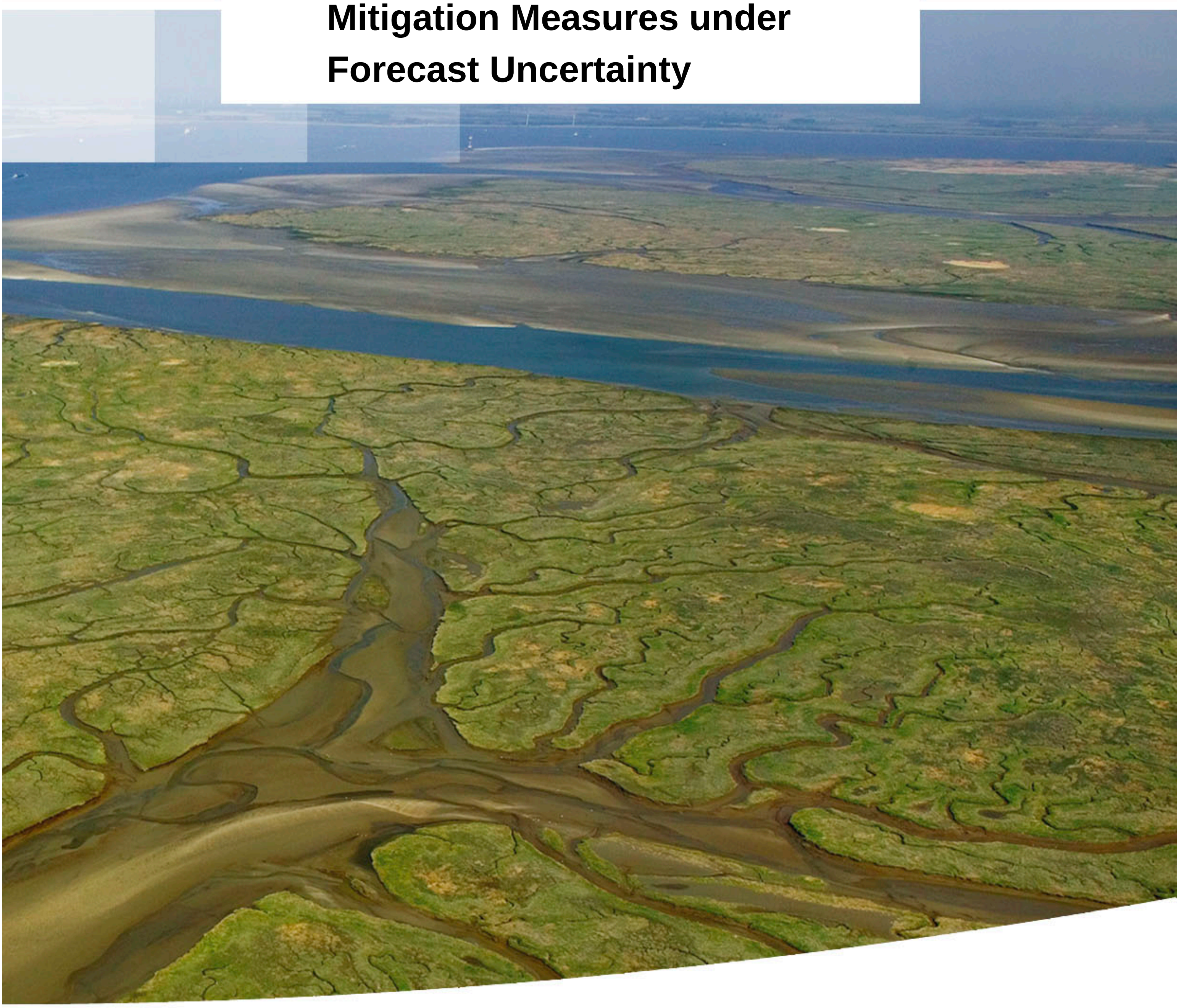
Deltares

Enabling Delta Life

Multi-stage Stochastic and Robust Optimization of Flood Mitigation Measures under Forecast Uncertainty



Multi-stage Stochastic and Robust Optimization of Flood Mitigation Measures under Forecast Uncertainty

Dirk Schwanenberg
Jorn Baayen
Matthijs den Toom
Peter Gijsbers

1221016-000

Deltares

Title
Multi-stage Stochastic and Robust Optimization of Flood Mitigation Measures under Forecast Uncertainty

Project	**Reference**	**Pages**
1221016-000	1221016-000-ZWS-0010	57

Keywords
Place keywords here

Summary
Inception Report of the TKI project "Multi-stage Stochastic and Robust Optimization of Flood Mitigation Measures under Forecast Uncertainty" summarizing the first 6 months of the project execution.

References
Place references here

Version	Date	Author	Initials	Review	Initials	Approval	Initials
	Aug. 2016	D. Schwanenberg		KJ van Heeringen		G. Blom	

State
final

Contents

1 Introduction

1.1 Background

Ensemble Streamflow Forecasting becomes a well-established technique in operational (flood) forecasting centers to assess forecast uncertainty. Currently, these forecasts are communicated to decision makers; however, taking decisions is still up to the subjective experience of the specific stakeholder. Due to the large amount of information in ensemble forecasts, this task is a major challenge in particular when time is limited during ongoing flood events.

There is a lack of objective methods to take qualified decisions under consideration of forecast uncertainty. Whereas stochastic optimization techniques based on ensemble forecasts are applied in other water management domains (e.g. for scheduling hydropower assets), they are so far not used in the scope of flood forecasting and early warning systems and comparable system for daily operations and droughts. One major reason is probably the conceptual difficulty to integrate binary decisions ("Evacuate a region or not") or logical constraints ("Measure A excludes measure B") into the decision-making under consideration of forecast uncertainty.

1.2 Objectives

This research will assess the application of several multi-stage stochastic and robust optimization approaches in combination with a mixed-logical, multi-objective optimization setup to model flood mitigation measures under forecast uncertainty. We will investigate the potential and applicability of these approaches to provide objective decision support to stakeholders in particular in the flood management domain. Where applicable, these approaches can also be applied in the general daily water management operations or for other purposes.

The conceptual assessment gets completed by a technical evaluation of in-house and external software packages with a focus on software features and software architecture. It should deliver a clear vision on the most suitable software framework for the requirements we identified for representative stakeholders.

1.3 Report Structure

Chapter 2 provides an inventory of short-term water management applications from the decision maker's perspective. An inventory of individual problem setups from relevant stakeholder leads to the definition of a number of representative test cases which serve as benchmarks in further research.

Chapter 3 repeats the inventory for the analyst, i.e. the tools which solve the test cases above to provide decision support for the decision maker. It treats the representation of the water system in a model, the deterministic optimization setup, the treatment of forecast uncertainty, the consideration of multiple objectives and finally the setup of a new conceptual and technical framework of a refactored or new software package.

Whereas chapters 2 and 3 document the inception phase (month 1-6), Chapter 4 includes the Statement of Work (SOW) for the remaining 30 months of the project.

2 Inventory (Decision Maker)

2.1 Stakeholders

2.1.1 Overview

This section summarizes the findings of a workshop for stakeholders held December 8, 2015 in Utrecht. The objective is to provide an inventory about decision support and real-time control applications for the short-term management of Dutch water systems. User stories present the motivation of the stakeholders to setup such systems, relevant processes in the water system from a management context as well as objectives and constraints of the management.

2.1.2 Rijkswaterstaat (RWS) / Ministry of Infrastructure and the Environment

User Story 1 (RWsOS Rivieren)
Stakeholder: RWS-WMCN (Eric Sprokkereef, Hendrik Buiteveld)

The operational forecasting system “RWsOS Rivieren” is operated by RWS-WMCN (Water-Management Centrum Nederland). It computes deterministic and probabilistic forecasts of water level and flow for the Rhine and Meuse river basins. Based on these forecasts and dedicated post-processing algorithms, RWS-WCMN supplies a validated deterministic forecast as basis for the decision making of several public stakeholders.

Stakeholders include the Dutch water boards which use the RWS-WMCN forecast as a boundary condition of their own forecasting and decision support systems both for the daily management of the water systems and its control during extreme events such as floods. Concerning the crisis management during flood events, all stakeholders in a region are organized in 25 so-called “Veiligheidsregios” which take over the coordination of crisis situation and the related decision making.

Although RWS-WMCN is not directly responsible for the decision-making, their forecasts and its related uncertainty is the most important building block for the downstream decision-making. Therefore, a better insight into the added value of this data is needed to supply the stakeholders with better products. As a representative case, we consider the implementation of an evacuation measure along the Meuse River. It requires a typical lead time of 3-4 days to get implemented. The benefit of such a measure results from the trade-off between the implementation costs of the evacuation, its damage reduction and other aspects such as the credibility of the forecaster (a high false alarm rate decreases the acceptance of a forecasts in the future).

Alternatively we may consider the case of closing the 'Ramspol stormvloedkering'. This decision to close this barrier is triggered by the storm surge (wind and wave conditions) on the IJssellake side but has negative implications for the ability to discharge fluvial water from the river IJssel and the river Vecht. A prolonged closure of this barrier will result in increased risk of upstream flooding due to limited drainage capabilities of the rivers. Main sources of uncertainty are:

- the meteorological (precipitation, wind), hydrodynamic (waves and water level rise) and hydrological (inflow boundary conditions) forecast uncertainty

User Story 2 (RWsOS IWP)
Stakeholder: RWS-WVL (Wim Werkman, Herbert Berger)

The operational forecasting and decision support system “RWsOS IWP” is developed and operated by RWS-WVL (Department Water, Traffic and Environment). It integrates and visualizes data of the Dutch national canal network (primarily water levels, flows and structural settings), collects or forecasts the fluxes of the water system and supplies decision-support tools for its daily operation. It is a country-wide, internal application of RWS. In its final extend, it will cover approximately 12-13 regional systems.

An appropriate process description of the canal system is achieved either by zero-dimensional models (storage nodes with interconnected structures) or one-dimensional hydraulic models including wind forces. Hydraulic structures should consider head-dependencies on their capacity and energy consumption. On top of the hydraulics, water quality processes such as contamination, salt intrusion and blue algae bloom can be relevant to the operation of such systems and should be considered in the decision-support.

A typical application in the RWsOS IWP has a forecast and control horizon of 2 days. It works with a time resolution of 10 min if tidal boundaries are present or on an hourly resolution otherwise. The control problem is multi-objective including water level setpoints for navigation, salt concentration, fish migration, organizational limitations and energy costs. The latter can be achieved by i) negotiating dedicated energy prices for example during day and night time, ii) requesting a daily energy consumption from a utility which schedules it within the day according to its highest financial benefit.

An important aspect of the control is the need to advice both on continuous (for example gate settings between min/max range with a maximum rate-of-change) and discrete decisions (a pump is either on or off). Furthermore, logical constraints should be considered. Examples include:

- minimum runtimes for a pump if switched on or off,
- no or only few changes at night (operator-friendly) if possible,
- the use of a structure only if the head is larger than a threshold.

Main sources of uncertainty in the operation of such system are:

- errors in the water balance of the canal system due to incomplete or erroneous observations (inflows and outflows, lock operations, etc.),
- the meteorological (precipitation, wind) and hydrological (inflow boundary conditions) forecast uncertainty,
- and the process description of water quality components.

These systems are typically operated by relatively low to medium educated technicians, who also have many other duties. An important aspect of the control is to make clear to them why a certain decision is taken. The control should not be a black box to them.

User Story 3 (RWsOS Waterbeheer)
Stakeholder: LCW, RWS-WVL (Wim Werkman, Bert Kort)

The institutes Alterra and Deltares, the Netherlands Environmental Assessment Agency (Planbureau voor de Leefomgeving, PBL) and RWS Waterdienst cooperate in the development of a integrated surface-groundwater model referred to as the Netherlands

Hydrological Instrument (NHI), with a country-wide called LHM. It consists of the models MODFLOW, MetaSWAP, DM and Mozart. The last two components represent the surface water and will get revised in 2016-2018.

The LHM is embedded into RWsOS Waterbeheer and runs daily. The execution of the overall model is slow, useful for an overview of the current and expected water system status but less suited for what-if scenario's to support operational decision making. Therefore, a quick-scan tool based on a multi-objective RTC-Tools model is being developed (pilot completed April 2016, final version later in the same year) to support the national water allocation during low flows and droughts according to given allocation priorities. An evaluation will assess, whether RTC-Tools could replace the DM and Mozart components in the NHI.

2.1.3 Waterschap Noorderzijlvest / Regional Water Authority Noorderzijlvest

User Story 4 (FEWS Noorderzijlvest)
Stakeholder: Waterschap Noorderzijlvest (Jan Gooijer, being replace by Arne Roelevink)

The regional water authority Noorderzijlvest (NZV) operates an operational forecasting and decision support system in the northeast of the Netherlands. It integrates and visualizes data of the regional water system, computes deterministic forecasts by hydrological and hydraulic models and implements the real-time control of the water system. A future use of probabilistic forecasts (GLAMEPS) is intended.

The required process description for the representation of the canal system is similar to the one in User Story 2 except for the need for modeling water quality processes (yet).

The objectives of the decision support component depend on the flow regime. During low and medium flows, the system conducts the daily water management under consideration of the main objectives of the fulfilment of water level setpoints and the cost-efficient operating of pumps with as little energy and energy costs as possible. The full automation of this mode as real-time controller is on the way putting an emphasis on the required robustness of the optimization approach. Other characteristics of the daily decision support are similar with the ones of User Story 2.

During flood events, the system serves as a decision-support component. The single objective of the system is the mitigation of flood peaks. Forecast horizon gets extended to 4-5 days. In this mode, the intended use of probabilistic forecasts in combination with a stochastic optimization procedure will increase the robustness of the suggested decisions. It will provide a better and more stable forecast and considers the forecast uncertainty explicitly in the decision-making procedure.

2.1.4 Water-Energy-eXchange (WEX)/JIP Slim Malen

Stakeholders: Waterschap Zuiderzeeland, Hollands Noorderkwartier, Friesland, Rivierenland, Brabantse Delta, Hollandse Delta, Rijnland, Scheldestromen, STOWA, Rijkswaterstaat, Eneco, Delta, Actility, Aliander/EnergieExchangeEnabllers, Xylem (Deltares: Ivo Pothof)

The project Water-Energy-eXchange (WEX)/JIP Slim Malen, also co-funded by TKI Deltatechnology, aims at the optimization of energy consumption and/or energy costs in regional Dutch water systems. Initially it considered 4 pilot studies for the regional water authorities Waterschap Zuiderzeeland, Hoogheemraadschap Hollands Noorderkwartier, Waterschap Rivierenland and Wetterskip Fryslân. By now many other waterboards have joined this project. The typical use case is a modified version of User Story 4 (FEWS NZV) with a number of essential differences.

First, the main goal of the decision support component during low and medium flows is optimization of energy consumption and/or energy cost. This implies that the water level set point will be no longer included in the objective function, but the water level range will be treated as a constraint only.

Secondly, the pump model will be extended in order to capture the correct relations between the pump discharge, pumping power and pump head over the allowable range of pump speeds, since many pumping stations have variable speed drive.

These differences with User Story 4 will allow for more accurate prediction of the power consumption of the pumping stations and the flexibility in the pumping station operation. In this way, the water boards will be able to sell the available flexibility at different energy markets, typically the day ahead market (APX).

2.1.5 Rekenen aan Slim Water Management

Stakeholders: Witteveen+Bos (Hoogheemraadschap Delfland), Nelen&Schuurmans (Hoogheemraadschap Hollands Noorderkwartier), Hoogheemraadschap Rijnland, Waterschap Noorderzijlvest (Floris Knot)

The TKI project “Rekenen aan Slim Water Management” aims at the application of innovative techniques for the smart water management in regional Dutch water systems. Besides the energy- and cost-efficient daily management, it also focuses on decision making during flood events.

In the project, user stories such as the ones above will be further refined and related implementation for the 4 water authorities will show the applicability of innovative techniques in a real-world setup.

2.1.6 Other Parties

Other activities with a relation to this project include:

- Ongoing assessment of the next generation RIBASIM software (MSc. Tiaravanni Hermawan under supervision of Peter Gijsbers and Eelco van Beek) to assess suitability of RTC-Tools for strategic planning of water resources and reservoir management
- Co-operation with Hoogheemraadschap Delfland (KlaasJan van Heeringen, Bart Dekens)
- Control of the heat network of TU Delft (contractors: Deltares, Kuijpers, Deerns, Priva BV): heat control in TUD buildings, current phase 2 (present – April 2016, reporting until September 2016) with feasibility study of predictive controller, potential SCADA system integration of predictive controller in phase 3

2.2 Selection of Representative Problem Setups

2.2.1 Overview

In this section, we define a number of representative test cases for further analysis based on the stakeholder inventory described above. Since this inventory is primarily conducted for typical Dutch water system and does not represent the full range of already existing RTC-Tools applications, we add a number of complementary cases to broaden the scope of the analysis. This includes the use of rainfall runoff and flow routing components within variational data assimilation applications and the short-term management of multi-purpose reservoir systems. The test cases are summarized in Table 2.1.

Table 2.1 Overview of representative problem setups

Case	Description	Application	Comments
HM-FR	Hydrological Modeling (Flow Routing) with various variable-parameter routing schemes	Hydrological flow routing as component in distributed hydrological models with variational data assimilation, flow routing between and downstream of reservoirs	Optimization variables << model states, therefore, preference for a sequential setup, but also need for collocated setup between reservoirs, 2nd-order derivatives required for collocated setup
RS	Reservoir System with multi-purpose reservoirs	Short-term optimization of the reservoir systems considering multiple objectives such as flood mitigation, hydropower generation, etc.	Optimization variables in the order of the model states, preference for collocated setup and 2nd-order derivatives, optional extension to hybrid systems and stochastic optimization, simple upstream to downstream routing
CS-CON	Canal System with Continuously Operated Structure(s)	Short-term optimization of a low-land water system as operated by Dutch water boards, relevant objectives include flood mitigation and cost-aware drainage	Comparable to case RS, but with more sophisticated flow processes (hydraulic routing), pumps instead of turbines, tidal boundaries, option for stochastic optimization
CS-DIS	Canal System with Barrier (Open / Closed)	According to CS-CON, but with discontinuous decisions, logical conditions etc.	According to CS-CON, but with dedicated mixed-integer optimization algorithms, option for stochastic optimization
CS-LES	Canal System with Lateral Extraction requests under Shortage conditions	Multi-objective water allocation	Priority based allocation using sequential goal programming optimization algorithm, option for weighting factor based LP approach
EV	Evacuation Measure Based on Uncertain Forecasts	Decision if and when an authority should initiate an evacuation measure	Application beyond the water system to address the impact of a forecast and its uncertainty on decision making

In the order of the cases, the hydrological modeling components above (HM-FR) are applied in particular within variational data assimilation methods (aka 4Dvar) to update model system states with observed historical data. The hydrological flow routing scheme (HM-FR) may also be used in combination with the reservoir systems model (RS). The latter is primarily applied to the short-term optimization of multi-purpose reservoir systems, but may also serve as a reservoir component within hydrological models. The main difference between the HM and RS cases is the ratio between optimization variables and model states as well as the need to constrain these states. Whereas the number of model states is significantly larger than the number of optimization variables in the HM cases, these are in the same dimension in the RS reservoir model. This favors a sequential[1] optimization setup for the HM cases and a collocated one for the RS case. Furthermore, the need to constrain states, for example the forebay elevation of a reservoir, is more relevant for the RS case and more easily implemented in the favored collocated setup.

The cases CS-CON and CS-DIS are derived from the inventory of the previous section. Both cover the short-term optimization of a typical Dutch canal or system. CS-CON covers only continuous control decisions, for example the control of a crest level in the range of minimum and maximum bounds. CS-DIS extends this case to binary decisions (“A pump is either on or off.”) and logical constraints (“If pump A runs, gate B cannot be used.”). One of the challenges in the WEX use case, is to find the most appropriate problem formulation, which might be a CS-CON formulation with the pumping station discharge as the main decision variable or a certain CS-DIS formulation. Case EV goes beyond the domain of the water system and assesses the application of decision support techniques to the implementation of an evacuation measure due to flooding.

All short-term control cases (RS, CS-CON, CS-DIS, CS-LES, EV) have multiple objectives and the stakeholders’ input is required as regards the trade-off of these objectives. Furthermore, all cases profit from the explicit consideration of forecast uncertainty in the decision making process to obtain more robust decisions.

[1] In a sequential optimization setup, only control variables become optimization variables. Model states such as the water level in a reservoir depend on these control variables. In contrary, the collocated setup handles both control variables and states as optimization variables and includes the process equations as equality constraints of the optimization problem. The pros and cons of each setup are problem dependent.

2.2.2 HM-FM: Hydrological Modeling (Flow Routing)

Process Description and Schematization

Hydrological routing schemes are frequently used in semi-distributed and distributed hydrological models and flow forecasting systems. They achieve accuracy close to full dynamic models for rivers with medium and steep slopes (without backwater effects), but have a much higher computational performance.

Various hydrological routing schemes can be formulated as a cascade of lumped nonlinear reservoirs (Schwanenberg & Alvarado Montero, 2016) according to the ordinary differential equation (ODE) given by

$$\frac{dS\left(I,Q,p\right)}{dt} - I + Q = 0 \tag{1.1}$$

where S is the storage, I and Q are the inflows and outflows of the reservoir, and p are parameters. A discrete-time form of Eq. (1.1) is achieved by an application of the θ-method to express the fluxes $I^{k-1/2}, Q^{k-1/2}$ as variables of the time steps $k-1, k$ by

$$\begin{aligned} I^{k-1/2} &= \left(1-\theta_I\right) I^{k-1} + \theta_I I^k \\ Q^{k-1/2} &= \left(1-\theta_Q\right) Q^{k-1} + \theta_Q Q^k \end{aligned} \tag{1.2}$$

to receive

$$\begin{aligned} F\left(I^{k-1,k}, Q^{k-1,k}\right) = & \frac{S^k\left(I^k, Q^k, p\right) - S^{k-1}\left(I^{k-1}, Q^{k-1}, p\right)}{\Delta t} \\ & -\left(1-\theta_I\right) I^{k-1} - \theta_I I^k + \left(1-\theta_Q\right) Q^{k-1} + \theta_Q Q^k = 0 \end{aligned} \tag{1.3}$$

where F is a function representing the mass error in the reservoir and θ_I, θ_Q are time weighting coefficients with unconditional stability in the range [0.5, 1]. Details of the numerical implementation in the existing RTC-Tools package are provided in Schwanenberg & Alvarado Montero (2016).

One application of the routing scheme is as a component of a hydrological model. In this case, model states significantly outnumber the optimization variables. This favours a sequential optimization setup. It requires a time integration by an iterative, reservoir-wise solution of Eq. (1.3). The computation of the 1st-order derivatives can be achieved by application of the implicit function theorem and the adjoint sensitivity equation (Alvarado-Montero et al., 2006).

An alternative implementation is a collocated setup by the introduction of Eq. (1.3) as an equality constraint of the optimization problem. This is the method of choice for example for routing reaches between two reservoirs in case of the short-term management of reservoir systems. It enables an efficient use of hard constraints on the forebay elevation of the reservoir and other quantities.

The test case considers the use of the routing model in a data assimilation setup. This can be described by a penalty on the deviation of the true state x and the simulated state m as well as the observed state o provided by

$$J = \int_t \frac{\left(x-m\right)^2}{\sigma_m^2} + \frac{\left(x-o\right)^2}{\sigma_o^2} dt \tag{1.4}$$

where σ_m, σ_o is the uncertainty, i.e. error variance, of the simulation and the observation, respectively. We introduce the state update Δx to receive the true state x from the simulated state m by $x = m + \Delta x$ and receive

$$J = \int_t w_m \Delta x^2 + w_o \left(x-o\right)^2 dt \tag{1.5}$$

where $w_m = 1/\sigma_m^2,\ w_o = 1/\sigma_o^2$ are referred to as weighting factor.

Test Case HM-FR1

Table 2.2 Test Case HM-FR1

Schematization	Muskingum-Cunge routing with a trapezoidal channel cross section used in Todini (2007) and Schwanenberg & Alvarado-Montero (2016): • $L = 100\,\mathrm{km}\quad \Delta x = 2\,\mathrm{km}$ • $\Delta t = 3600\,\mathrm{s}$ • Trapezoidal cross section with $B_0 = 15\,\mathrm{m}$, slope of 1:5 • Bed slope of 0.00025, Manning roughness $n = 0.035$
Simulation Period Boundary Conditions	Simulation period is $T = [0,100\,\mathrm{h}]$ The inflow hydrograph is adopted from experiments in NERC (1975), Todini (2007) and others: $$Q_I(t) = Q_{base} + \left(Q_{peak} - Q_{base}\right)\left[\frac{t}{T_p}\exp\left(1 - \frac{t}{T_p}\right)\right]^{\beta}$$ where $\beta = 16$, $Q_{peak} = 900\,\mathrm{m^3 s^{-1}}$, $Q_{base} = 100\,\mathrm{m^3 s^{-1}}$ and $T_p = 24\,\mathrm{h}$.
Optimization Setup	We assume an observed outflow Q_O at the downstream boundary according to the equation of the inflow hydrograph above with the modified parameters $Q_{peak} = 700\,\mathrm{m^3 s^{-1}}$ and $T_p = 38\,\mathrm{h}$. The objective functions penalizes updates of the inflow boundary as well as deviation between observed and simulated (and updated) flow at the downstream boundary: $$\min_{\Delta Q_I} \sum_k w_n \Delta Q_I^{\,2} + w_d\left(Q_{O,sim} - Q_{O,obs}\right)^2$$ where ΔQ_I is the update (optimization variable), $Q_{O,obs}, Q_{O,sim}$ are observed and simulated outflows, $w_n = 0.1$ and $w_d = 1$ are weighting factors.
Validation	Criteria of the validation are: • Convergence history of the optimization • CPU time per iteration

2.2.3 RS: Reservoir System

The planning and management of hydropower reservoir systems constitutes a highly relevant case for model predictive control. Each reservoir in the system needs to be operated in such a way that the system wide objectives are met, given the predicted uncertain inflows and system load. These objectives may relate to various strategies and/or obligations of the hydropower operator, and are often conflicting.

Process Description and Schematization

A reservoir system is schematized by the following main components:
1 hydropower reservoir,
2 routing reach between two reservoirs or downstream.

In this case, the reservoir component is a compact representation of a hydropower facility, based on aggregated turbine and spillway characteristics. The component implements volume conservation in the reservoir by

$$\frac{dS}{dt} = Q_I - Q_O \tag{1.6}$$

where S is the reservoir storage, Q_I, Q_O are the total inflows into the reservoir and its outflow. The outflow consists of turbine flow Q_T and spillage Q_S according to

$$Q_O = Q_T + Q_S \tag{1.7}$$

The level-storage relation defines the dependency of the storage on the forebay elevation z_{fb}

$$S = f\left(z_{fb}\right) \tag{1.8}$$

Furthermore, the power generation is computed by

$$h = z_{fb} - z_{tw} \tag{1.9}$$

$$P = \eta\, \rho\, g\, h Q_T \tag{1.10}$$

where z_{tw} is the tailwater elevation, h is head, P is the power generation and $\eta, \rho (= 1000\,\mathrm{kg/m^3}), g\,(= 9.81\mathrm{m/s^2})$ are coefficient for the turbine efficiency, the density of water and the acceleration due to gravity, respectively.

For simplicity, the test case considers a linear level storage relation according to

$$S = S_0 + Az_{fb} \tag{1.10}$$

where S_0 is a reference storage at $z_{fb} = 0$ and A is the reservoir surface area. In addition, the tailwater elevation z_{tw} and turbine efficiency η are assumed to be reservoir-specific constants.

The routing element is modeled by a simple delay term. The downstream discharge Q_D is equal to delayed upstream discharge Q_U according to

$$Q_D\left(t\right)=Q_U\left(t-\tau\right) \tag{1.10}$$

where τ represents the time delay.

A reservoir system constitutes of linked routing elements and reservoir components, forming a network. The reservoir inflow Q_I is equal to externally specified inflows Q_E, added with the sum of the routed outflows from the connected upstream reservoirs Q_D by

$$Q_I = Q_E + \sum_{upstream} Q_D \tag{1.11}$$

For each of the most upstream reservoirs in the network the externally specified inflow (Q_E) is the only source of water. For the other reservoirs, this term represents the accumulation of lateral inflows along the upstream river reaches.

In the test case, we consider a reservoir system consisting of three reservoirs and two connecting routing elements. Table 2.1 below lists the components in order from upstream to downstream. The hydrographs for the external inflows are adopted from case HM-FR1:

$$Q_E\left(t\right)=Q_{base}+\left(Q_{peak}-Q_{base}\right)\left[\frac{t}{T_p}\exp\left(1-\frac{t}{T_p}\right)\right]^{\beta} \tag{1.11}$$

with parameters as specified in Table 2.3. The simulation period T = [0, 30d].

Table 2.3 Schematization for RS test cases, listing the components in order from upstream to downstream.

Component ID	**Type**	**Characteristics and forcing**
RES_UP	Reservoir	A = 5.0 × 10^7 m^2 z_{tw} = 500 m η = 0.88 Q_{base} = 150 m^3/s Q_{peak} = 750 m^3/s T_p = 6d β = 3
UP_TO_MID	Routing	τ = 12 hr
RES_MID	Reservoir	A = 5.0 × 10^8 m^2 z_{tw} = 250 m $Q_S = Q_O$, i.e. no power generation Q_{base} = 300 m^3/s Q_{peak} = 1500 m^3/s T_p = 6d

		β = 3
MID_TO_DOWN	Routing	τ = 24 hr
RES_DOWN	Reservoir	A = 5.0 × 10^8 m^2 z_{tw} = 0 m η = 0.88 Q_{base} = 150 m^3/s Q_{peak} = 1500 m^3/s T_p = 6d β = 3

Test Case RS1

In this test case, a set of constraints is imposed that reflects common operational requirements. The constraints may reflect facility characteristics (e.g. turbine capacity), environmental obligations (e.g. restricted forebay operating range), and the strategic use of water resources (e.g. no spill in the dry season). For each of the three reservoirs, forebay elevation is restricted according to

$$z_{fb,\min} \leq z_{fb} \leq z_{fb,\max} \tag{1.12}$$

In addition, the two facilities with generation (RES_UP and RES_DOWN) need to satisfy

$$0 \leq Q_T \leq Q_{T,\max} \tag{1.13}$$

$$0 \leq Q_S \tag{1.14}$$

$$0 \leq P \leq P_{\max} \tag{1.15}$$

Finally, rate-of-change and average constraints for the total outflow are imposed to ensure smooth control by

$$dQ_{O,\min} \leq \frac{dQ_O}{dt} \leq dQ_{O,\max} \tag{1.16}$$

$$\bar{Q}_{O,\min} \leq \int_{t1}^{t2} Q_O \leq \bar{Q}_{O,\max} \tag{1.17}$$

The objective considers the maximize of the total generation over the forecast horizon by

$$\max \int_t \sum_{projects} P\, dt \tag{1.17}$$

In order to prevent the reservoirs from drafting towards the end of the simulation horizon, the forebay elevation z_{fb} at $t = T$ is fixed by

$$z_{fb}\left(t = T\right) = z_{fb}^T \tag{1.18}$$

Table 2.4 Reservoir characteristics

	Component ID		
	RES_UP	**RES_MID**	**RES_DOWN**
Forebay	[580, 650] m	[300, 303] m	[5.5, 10] m
Outflow rate-of-change	[-50, 50] m^3/s/h	[-300, 300] m^3/s/h	[-150, 150] m^3/s/h
Outflow average over sliding week	[0, 250] m^3/s	[0, 1500] m^3/s	[0, 750] m^3/s
Turbine	[0, 300] m^3/s		[0, 1000] m^3/s
Generation	[0, 300] MW		[0, 55] MW

Test Case RS2 (discontinuous spillage)

Test case RS1 constitutes a fully continuous problem formulation. In this test case, discrete behavior is introduced by replacing the zero spill flow constraint for RES_UP by a constraint of the form

$$0 \le Q_S \le a \;\; or \;\; b \le Q_S \tag{1.19}$$

where the parameters $0 < a < b$ define a forbidden spill range in the interval $[a,b]$.

Test Case RS3 (unit dispatch)

This test case constitutes another extension of test case RS1. Like in test case RS2, discrete behavior is introduced, but now by allowing individual turbine units to be switched on and off, and penalizing such switches in the objective function. Instead of the approach based on aggregated facility characteristics, the total turbine flow and power generation become a sum of their individual units

$$Q_T = \sum_i Q_{Ti} \tag{1.20}$$

$$P = \sum_i P_i \tag{1.21}$$

Furthermore, the turbine efficiency of a unit gets dependent on the flow and head through this unit according to

$$\eta = f\left(Q_{Ti}, h\right) \tag{1.22}$$

The turbine efficiency η has either a piecewise linear relation with turbine flow Q_T and head h. Alternatively, the relation can be fitted by a polynomial.

The set of constraints is augmented with constraints on the turbine flow of the individual units, which may be switched on in a certain range or off,

$$Q_{Ti} = 0 \;\; or \;\; Q_{Ti,\min} \le Q \le Q_{Ti,\max} \tag{1.23}$$

The objective additionally takes into account the costs for the start-up and shut-down of a turbine by

$$\max \int_t \sum_{projects} P\,dt - \sum_{switches} \gamma \tag{1.24}$$

where γ represents the (scaled) cost for switching a turbine on or off.

2.2.4 CS-CON: Canal System with Continuously Operated Structure(s)

Process Description and Schematization

An appropriate process description of water levels and flows in low-land canal networks within water management applications are the 1D shallow water or Saint Venant equations. In its non-conservative form, they read

$$\frac{\partial A}{\partial t}+\frac{\partial Q}{\partial x}=0 \tag{1.25}$$

$$\frac{\partial Q}{\partial t}+\frac{\partial}{\partial x}\left(\frac{Q^2}{A}\right)=gA\left(-\frac{\partial z}{\partial x}-S_f\right) \tag{1.26}$$

where A is the flow cross section, Q is the flow, z is the water level above reference datum, S_f represents friction slope in the dimensions time t and space x, and g is the acceleration due to gravity. A beneficial simplification of these equations for water resources applications on courser computational grids is the inertial wave model we achieve by neglecting the convective acceleration term in Equation (1.26) to receive

$$\frac{\partial Q}{\partial t}=gA\left(-\frac{\partial z}{\partial x}-S_f\right) \tag{1.27}$$

The set of Equations (1.25) and (1.27) can be schematized on a staggered grid leading to a system of Ordinary Differential Equations (ODEs) and Algebraic Equations (AE) according to

$$\frac{dS_i(z_i)}{dt}=\sum_j Q_j \tag{1.28}$$

$$\frac{dQ_j}{dt}=gA_j\left(-\frac{z_u-z_d}{\Delta x}-S_{f,j}\right) \tag{1.29}$$

$$S_i=f\left(z_i\right),\ A=g\left(z_j\right),\ S_{f,j}=h\left(Q_j,z_j\right) \tag{1.30}$$

where the indices i, j represent the discrete locations for storage and flow, respectively, z_u, z_d refer to the water level in the upstream and downstream storage nodes from a branch perspective, the functions $f(), g(), h()$ represent the level-storage relation of a node, the flow cross section and the friction losses at the branch, respectively. The water level at a branch can be expressed either by a central or upwind formulation provided by

$$z_{j,central}=\frac{1}{2}\left(z_u+z_d\right),\ z_{j,\mathrm{upwind}}=z_u \tag{1.31}$$

In Dutch canal system with relatively constant water levels, the central formulation is preferred due to its higher accuracy. In steeper river reaches with higher Froude numbers, the upwind formulation can be advantageous because of its higher numerical robustness. The friction slope can be expressed as

$$S_{f,j} = \frac{Q_j \left| Q_j \right|}{C^2 A_j^2 m_j} \tag{1.32}$$

where C is the Chézy roughness coefficient and m is the hydraulic radius. Alternative formulations of the friction slope, for example the empirical Manning formula, can be implemented by expressing the Chézy coefficient by a Manning coefficient n according to

$$C = \frac{m^{1/6}}{n} \tag{1.33}$$

The test case considers an inland water system according to Figure 2.1.

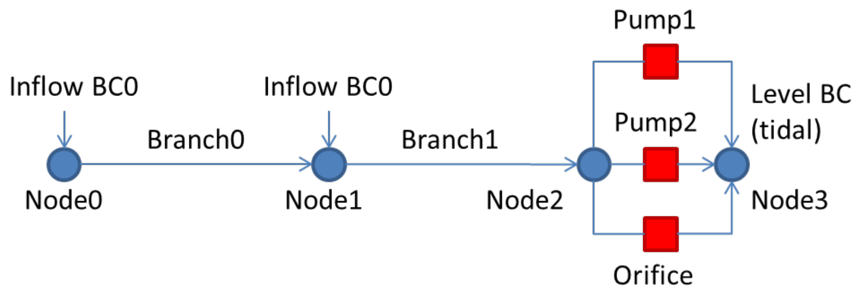


Figure 2.1 Schematization of the canal system in test case CS-CON

Nodes 0-1 represent the inland canal system. Inflow from the surrounding region enters the system in the nodes 0 and 1 by inflow boundary conditions. The nodes are connected by two flow branches according to the inertial wave model of Equation (1.27). This implies that the flow direction in branch 1 may change direction depending on the inflow boundary conditions and the drainage strategy at node 2.

Node 3 represents the sea by a water level boundary condition and the related tidal signal. The drainage of the canal is achieved by two pumps and an orifice. Pump 1 is an electric pump with an installed capacity of 5 m^3/s. Pump 2 is a diesel-powered pump of a capacity of 10 m^3/s. We assume both capacities head-independent in the first three cases, then head-dependent. The orifice can release water by gravity flow, if the upstream water level at node 2 is higher than the downstream water level at node 3. A return flow of salt water from node 3 to node 2 is not permitted.

The flow capacity of the orifice is given by

$$Q = \begin{cases} 0 & \text{if } h_{down} > h_{up} \\ w_s \mu d_g \sqrt{2g\left(h_{up} - \left(z_s + \mu d_g\right)\right)}, & \text{if } h_{down} < z_s + d_g \text{ (free flow)} \\ w_s \mu d_g \sqrt{2g\left(h_{up} - h_{down}\right)} & \text{otherwise (submerged flow)} \end{cases} \tag{1.34}$$

where w_s is the gate width, μ is the contraction coefficient, d_g is the gate opening, z_s is the crest level of the gate, and h_{up}, h_{down} are the upstream and downstream water levels, respectively. Depending on the definition of the parameter μ (a typical value is around 0.67), the capacity has a discontinuity in the transition from free to submerged flow.

Test Case CS-CON1 (energy reduction)

Table 2.5 Test Case CS-CON1

Schematization	Nodes 0-3 have a constant water surface of $A = 100,000\,\mathrm{m}^2$ and a level-storage relation of $S = A(z + 2\,\mathrm{m})$. The canal reaches of branch 0 and 1 have the following characteristics: • $L = 10\,\mathrm{km}$ • Trapezoidal cross section with $B_0 = 5\,\mathrm{m}$ at $z = -2\,\mathrm{m}$, slope of $1:2$ • Bed slope of 0, Chezy roughness $C = 30$ The orifice has the characteristics: width $w_s = 10\,\mathrm{m}$, contraction coefficient $\mu = 0.67$ and crest level $z_s = -1.8\,\mathrm{m}$. The system is schematized with a time step of $\Delta t = 300\,\mathrm{s}$.
Simulation Period Boundary Conditions	Simulation period is $T = [0, 48\mathrm{h}]$ The inflow hydrographs are adopted from case HM-FR1 $$Q_I(t) = Q_{base} + \left(Q_{peak} - Q_{base}\right)\left[\frac{t}{T_p}\exp\left(1 - \frac{t}{T_p}\right)\right]^{\beta}$$ with the branch-dependent parameters: • Node 0: $\beta = 16$, $Q_{peak} = -10\,\mathrm{m}^3\mathrm{s}^{-1}$, $Q_{base} = 5\,\mathrm{m}^3\mathrm{s}^{-1}$ and $T_p = 24\,\mathrm{h}$ • Node 1: $\beta = 16$, $Q_{peak} = 40\,\mathrm{m}^3\mathrm{s}^{-1}$, $Q_{base} = 5\,\mathrm{m}^3\mathrm{s}^{-1}$ and $T_p = 12\,\mathrm{h}$ The tidal level boundary conditions is provided by $$z_{tidal}(t) = z_{mean} + \Delta z \sin\left(\frac{t\pi}{6\mathrm{h}}\right)$$ where $z_{mean} = -0.5\,\mathrm{m}$ and $\Delta z = 1\,\mathrm{m}$.
Optimization Setup	The objective function penalizes the use of pumps (and related energy consumption) as well as the deviation of a water level setpoint at node 3. $$\min_{Q_P, d_g} \sum_k w_p Q_p + w_{sp}\left(z_{2,sim} - z_{2,sp}\right)^2$$ where Q_p is the pump discharge (optimization variable), d_g is the opening height of the orifice (optimization variable), $z_{2sim}, z_{2,sp} = -0.8\,\mathrm{m}$ is the simulated water level at node 2 and its

	setpoint, respectively, $w_p = 0.1$ and $w_{sp} = 1$ are weighting factors for the use of pumps and the setpoint deviation. Furthermore, the use of the hydraulic structures is constraint by its physical bound given by $0 \le Q_p \le 15\,\mathrm{m^3/s}$ $0 \le d_g \le 1.0\,\mathrm{m}$ The water levels at nodes 0-2 is hard-constraint by a minimum water level according to $-1.0\,\mathrm{m} \le d_g$
Validation	Criteria of the validation are: • Convergence history of the optimization • CPU time per iteration

<u>Test Case CS-CON2 (energy cost reduction)</u>

Whereas we try to minimize the pump energy in case CS-CON1, this case considers the minimization of energy costs versus the deviation from water level setpoints. Modification related to case CS-CON1 are only present in the optimization setup

Optimization Setup	In comparison to case CS-CON1, we consider the pumps separately and penalize its use individually by considering a time-dependent weighting factor which represents the instantaneous energy price. $\min_{Q_{p1},Q_{p2},d_g} \sum_k w_{p1}^k Q_{p1} + w_{p2}^k Q_{p2} + w_{sp}\left(z_{3,sim} - z_{3,sp}\right)^2$ Both pumps get constrained separately by $0 \le Q_{p1} \le 5\,\mathrm{m^3/s}$ $0 \le Q_{p2} \le 10\,\mathrm{m^3/s}$ Optimization variables are the individual pump discharges Q_{p1}, Q_{p2} and the opening height of the orifice d_g .

<u>Test Case CS-CON3 (trade-off analysis)</u>

Up to this point, the two objectives pumping energy / costs and level setpoints in test cases CS-CON1 and CS-CON2 have been combined by a scalar weighting function. The main criticism of this technique from a user's perspective is the difficult interpretability of the impact of weighting factor changes on the trade-off between the objectives. The focus of this test case is to explore different techniques to define, assess and visualize the trade-off between the two objectives above to a user.

This covers:

- The assessment of the pareto front of pumping energy / costs and the setpoint deviation in combination with the scalar weighting function approach to quantify and visualize the trade-off between the objectives.
- The application of alternative multi-objective optimization approaches to find more user-friendly ways for the system operator to define his preferences by priorities:
    - Put priority on decreasing the maximum water level regardless of pumping costs until an acceptable threshold is reached.
    - Use remaining flexibility in the water system to reduce pumping costs in combination with a reasonable bandwidth of setpoint deviations.

The further enhancement of this case will closely depend on the numerical techniques addressed in the next chapter.

Test Case CS-CON4 (operational flexibility for energy utility)

A promising future business model for regional water boards is the operation of pumps in collaboration with energy utilities. In this setup, the water board requests its predicted daily energy demand for the pumps and the utility schedules this demand dependent on balancing requirements and the energy price. The energy consumption will vary depending on the starting time of pumps, since head will vary due to tidal dynamics. Hence the water board should provide information to the utility on this flexibility and associated extra pumping power requirement. Further the water level set-point should be shifted from the goal function to a boundary condition in this test case.

The operation of the water system inherits two main sources of uncertainty, namely
- the forecast uncertainty in the inflow prediction of the water system and
- the scheduling of the utility.

Both can be represented in a probabilistic ensemble.

The purpose of the test case is the setup of a stochastic optimization procedure to schedule the daily demand of the water authority in such a way, that
- the overall use of energy / energy costs is minimized,
- the setpoint deviations are minimized,
- the chance for unscheduled use of pumps (and its related penalty) is minimized.

The further enhancement of this case will closely depend on the numerical techniques addressed in the next chapter.

Test Case CS-CON5 (forecast uncertainty during flood events)

Similar to the base case CS-CON5, but with a forecast horizon of 5 days and a probabilistic forecast and a focus on flood mitigation

2.2.5 CS-DIS: Canal System with Barrier (Open / Closed)

Process Description and Schematization

The process description as well as the schematization of the water system is identical with the one in the CS-CON case.

Test Case CS-DIS2 (energy cost reduction)

This test case is based on case CS-CON2 with the following modifications:

- Both pumps can be only on or off. The operation of the orifice stays either continuous or gets replaced by a number of discrete settings.
- If a pump is switched on or off, it is required to stay in this position for at least 2 hours.

Test Case CS-DIS2b (energy cost reduction)

This case is identical to case CS-DIS2 with the difference of fixed steps for the pumps discharge.

Test Case CS-DIS3 (trade-off analysis)
Test Case CS-DIS4 (operational flexibility for energy utility)
Test Case CS-DIS5 (forecast uncertainty during flood events)

The test cases are based on cases CS-CON3 - CS-CON5 with the modifications above.

2.2.6 CS-LES: Canal System with Lateral Extraction requests under Shortage conditions

Process Description and Schematization

Water allocation problems generally are represented by a network of connected water balance elements with laterals to accommodate extractions. Objectives are the request for lateral extraction or instream flows. Control variables are the allocated extractions and allocated flows in the network. Under shortage conditions, not all requests can be met. This multi-objective problem can be formulated in a priority based approach using sequential goal programming techniques (see section 3.4.3) where all requests (i.e. objectives *i*) have been assigned a priority or goal order. The algorithm sequentially solves all objectives by priority order.

Test Case: CS-LES (priority based)

Test case considers a simple situation where the a channel has two lateral intakes (*QLat2* and *QLat3*). The channel itself has a downstream flow request (*Qin1*).

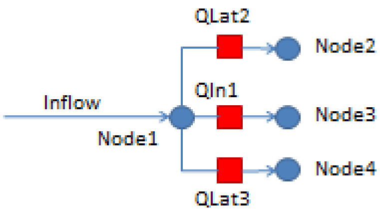


Table 2.6 Test Case CS-LES1

Schematization	• QIn_1 = 3, priority = 1 • $QLat_2$ = 10, priority = 2 • $QLat_3$ = 5, priority 3
Simulation Period Boundary Conditions	Simulation period is $T = \left[0, 48\text{h}\right]$ Q_{inflow} = variable between 2 and 20
Optimization Setup	The optimum solution is attained when the sum of deviation ($f_k(x)$) from all variable in all-time series reaches the minimum value. Nevertheless, the optimum solution must be inside the bounds of the inviolable hard constraints($g(x)$). It is also important to note that the sum of deviations of the higher goals $\left(f_i(x_{opt,i})\right)$ must remain constant or smaller after the lower goals ($f_i(x)$) are solved. $\min_x \left[f_1(x), f_2(x), \ldots, f_i(x) \right]$ subject to $g(x) \leq 0$ $\boldsymbol{f_i}(\boldsymbol{x}) = \varepsilon_i \ \forall i < k \qquad \varepsilon_{\boldsymbol{i}} = \boldsymbol{f_i}\left(\boldsymbol{x_{opt,i}}\right)$
Validation	Criteria of the validation are: • Allocation logic given order of priorities • CPU time per iteration

2.2.7 EV: Evacuation Measure Based on Uncertain Forecast

Process Description and Schematization

We assess the implementation of an evacuation measure with a required minimum lead time of 4 days. In a simple deterministic setup which serves as a reference, it is triggered by the up-crossing of a forecasted flow above a threshold at this specific lead time. It neglects forecast uncertainty both in the meteorological and hydrological forecast. In the corresponding probabilistic setup, we make use of a probabilistic forecast in the form of an ensemble or as a probability density distribution.

The costs of the evacuation measure are provided by

$$C(\delta,t)=\delta\left[C_{base}+c\left(t-t_{\min}\right)\right] \tag{1.35}$$

where $C(t)$ are the costs of the evacuation measure initiated at lead time t, C_{base} are the costs initiated at the minimum lead time $t_{\min}$, c is a discount on the costs of the evacuation measure when getting implemented with a larger lead time, δ is a binary variable which indicates if the evacuation measure is implemented (in case of $\delta=1$).

The flood damage is reduced by the evacuation measure by

$$D\left(\delta,t,Q_{\max}\right)=\begin{cases} D_{base}+d\left(Q_{\max}-Q_{th}\right), & \text{if } Q_{\max}>Q_{th}\wedge\delta=1\wedge t<t\left(Q_{th}\right) \\ 0 & \text{otherwise} \end{cases} \tag{1.36}$$

where $D\left(Q_{\max}\right)$ is the peak flow dependent damage reduction, D_{base} is the damage reduction at the inundation threshold Q_{th} by the evacuation measure, and d is the increase of the damage reduction for an increasing peak flow. This means that the evacuation measure has only a positive impact i) if the peak flow $Q_{\max}$ is above the inundation threshold Q_{th} (no damage reduction without damage and damage appears only when the threshold is up-crossed), ii) the evacuation measure is implemented and iii) it is implemented before the inundation threshold is reached.

Test Case EV1

Table 2.7 Test Case EV1

Schematization	Parameters of evacuation costs and damage reduction functions: • C_{base} = 1 M€, c = -100 k€/d, $t_{\min}$ = 96h • D_{base} = 10 M€, d = 1 M€/d • $Q_{threshold}=TBD$
Simulation Period	The deterministic forecast has a lead time of 10 days, the probabilistic one has a lead time of 15 days.

Boundary Conditions	An archive of actual deterministic and probabilistic flow forecasts is provided from the operational flood forecasting system of RWS.
Optimization Setup	The objectives functions in its deterministic version represents the overall benefit of the decision by $$\min_{\delta,t} C(t) - D(Q_{\max})$$ and can be extended to a stochastic optimization by a probability weighted sum according to $$\min_{\delta,t} \sum_{j=1}^{n} p_j \left[C_j(t) - D_j(Q_{j,\max}) \right]$$ Optimization variables are the binary decisions to evacuate or not and when.
Validation	Criteria of the validation are: • Performance of the probabilistic / stochastic setup in comparison with the deterministic one in terms of the o benefit of the decision J o false alarm rate of the evacuation measure

This case is subject to a sensitivity study on the following parameters:
- ratio between the costs for the evacuation measure and its damage reduction D_{base} / C_{base}
- discount on the evacuation measure c
- increase rate of the damage reduction d

3 Inventory (Analyst)

3.1 Model Library

3.1.1 Overview

In the context of water resources related real-time control and decision support, the main purpose of a model library is to provide simulation components for the prediction of processes and its optimization. This includes:

1. the availability of core library components for the most frequently used process models such as reservoirs and hydraulic cannel networks, related tests and application examples,
2. the easy extensibility of the library to dedicated processes,
3. configurable schematization options related to the time integration (explicit or implicit schema) and the level of collocation (collocated, multiple shooting or single shooting),
4. the robust and simple integration of the model library in optimization algorithms (for example by supplying derivatives or a symbolic model representation),
5. an easy and if possible GUI-supported model setup procedure.

3.1.2 RTC-Tools

The existing version of RTC-Tools 1.X supplies a model library on the level of C++ classes and configuration by XML. It includes a number of frequently used components such as several reservoir models and simplified hydraulic models, but also incorporates a number of dedicated, custom made components.

The integration of new components require:

- the implementation of a C++ class including functions for simulation and an adjoint mode (this means that the 1st order derivative needs to be coded by the developer)
- the implementation of a XSD schema for the configuration of the component as well as a data binding components to call the class constructor by the XML-embedded information.

The advantage of the approach is the high flexibility to implement arbitrary models in combination with any dedicated schematization. On the other hand, it implies a number of disadvantages such as:

- the error-prone manual implementation of the 1st-order derivative and the lack of higher-order derivatives,
- a relatively high effort to implement new components and the incapability to isolate dedicated, custom-made components from the core components and
- the lack of GUI-supported setup tools.

3.1.3 Modelling Languages (Modelica / JModelica)

Traditional languages for formulating generic optimization problems are AMPL and GAMS. AMPL originated in the 1980s, whereas GAMS dates as far back as the late 1970s. Having arisen well before the maturing of the ideas of object oriented software development in the

1990s, AMPL and GAMS require the specification of the combined model and optimization problem in a single, monolithic file, in which the physics, model, discretization, and optimization problem are all jumbled together.

Starting from the late 1990s, a group of researchers from the University of Lund in Sweden started to work on the object-oriented language Modelica for writing down mathematical models as differential equations. Using Modelica, one writes an ODE as such, instead of writing down a discretized solution. This separation of model and discretization leads to lean models. Furthermore, the models are kept well-structured by separating different model components into their own object classes.

Modelica compilers take a model - which may contain a hierarchy of child models - and flatten all the contained equations into a single system of equations. This single system of ordinary differential and algebraic equations may then be imported into a simulation or optimization engine. The open source JModelica.org compiler, for instance, compiles a Modelica model directly into an in-memory symbolic representation, available through an API.

The Modelica standard is open, and consequently many software packages are compatible with it. Modelica is now in heavy industrial use around the world at corporations such as Airbus, BMW, but also at DHI in the water sector, where it is used as the engine for MIKE WEST.

Modelica has been primarily used to model technical applications and did not receive a lot of attention in the hydrological community or for the modeling of environmental systems. The application of the framework to this kind of applications will be subject to further analysis in this project. The standard does not address schematization options, however, various time integrators for Modelica models are widely available in tools implementing the Modelica standard. Partial differential equations (PDEs) are not supported at this point and its consideration depends on a reformulation of the PDE as a system of ODEs.

3.1.4 Algorithmic Differentiation (CasADi)

For numerical optimization, the availability of first and second order derivatives of the objective function and constraints are beneficial and increase the efficiency of the optimization algorithm. Consequently, first and second order derivatives of the underlying system model are also desirable. Algorithmic Differentiation (AD) refers to a technique which supplies the derivatives of a given function.

The simplest automatic differentiation technique is to use *operator overloading*, i.e., to override the definitions of the mathematical operators (multiplication, additional, exponentiation, and so forth). The automatic differentiation operators don’t only compute the result of the operation, but also apply the rules of calculus to a second value representing the numerical value of the derivative. The advantage of this technique is its simplicity, whereas its principal downside is that the compiler is unable to optimize the derivative computations.

An alternative technique is the conversion of the source code of a function to code that computes the function as well as its derivatives. This technique, known as *source code transformation*, is not always practical in that it requires all functions that will ever be required for the optimization to be known at compile time.

The in-memory symbolic representation is done in the open source software library CasADi. It implements a variation of the source code transformation technique. In CasADi, one constructs a symbolic representation of one's mathematical function. The resulting symbolic function can be differentiated - as a whole - an arbitrary number of times, resulting in new symbolic functions. The original symbolic function may then be merged with its derivative functions, in a way that common sub-expressions, such as evaluations of trigonometric functions and logarithms, are shared. Finally, the combined symbolic function can be written to a C file, compiled using an optimizing C compiler, and loaded back as a shared library. The resulting C function provides accurate derivatives at high speed.

The JModelica.org framework provides a compiler back end that compiles Modelica models to in-memory symbolic CasADi representations. Furthermore, CasADi may supply the framework to re-implement the RTC 1.X model library and merge it with models originating from Modelica.

3.2 Deterministic Optimization Setup

3.2.1 Overview

The continuous form of an arbitrary process model can be either a system of ordinary differential equations (ODEs), differential-algebraic equations (DAEs) or partial differential equations (PDEs). Under the assumption that its discrete-time version is schematized by a single step method, it reads

$$x^k = f\left(x^{k-1}, x^k, d^k, u^k\right) \tag{2.1}$$

$$y^k = g\left(x^k, d^k, u^k\right) \tag{2.2}$$

where x, y, d, u are vectors for the states, model outputs, disturbance (forcing) and the control input, respectively, and the index t denotes the time index. Depending on the existence of the term x^k on the right-hand side of Eq. (2.1), the equation becomes either an explicit or implicit function. The latter requires an iterative solution.

The deterministic version of the optimum control problem can be defined as

$$\begin{gathered}\min_{u, x_*} \sum_{k=1}^{T} J_k\left(x, y, d, u\right) \\ h\left(x^k, y^k, d^k, u^k\right) \leq 0 \\ x_*^k - f\left(x_*^{k-1}, d^k, u^k\right) = 0\end{gathered} \tag{2.3}$$

where J is the objective function, h is a set of inequality constraints and x_* is the subset of states which become independent optimization variables and which corresponding state equations become equality constraints of the optimization problem.

3.2.2 Linearization Options

The most generic setup of the optimization problem above is achieved by the assumption of a nonlinear process model in combination with arbitrary constraints. However, many problems in water resources applications may be either linear or only slightly non-linear. In this case, a linearization of remaining nonlinear components leads to linear optimization model, for which dedicated optimization algorithms are available with significant better performance than the one of general nonlinear problems. The same aspect will hold for mixed integer (MI) problems.

One of the most frequently met nonlinearities in water resources models is the power generation of a turbine (or power consumption of a pump) according to

$$P = \eta \, \rho \, g \, h Q_T \tag{2.4}$$

where P is the power generation, η is the turbine efficiency (potentially head and flow dependent), ρ is the water density, g is the acceleration due to gravity, h is head and Q_T is the turbine flow.

A typical approach for the linearization of Eq. (2.4) is its conversion into a separable model. The introduction of the auxiliary variables u_1, u_2 according to

$$\begin{aligned} u_1 &= \frac{1}{2}\left(h + Q_T\right) \\ u_2 &= \frac{1}{2}\left(h - Q_T\right) \end{aligned} \tag{2.5}$$

where u_2 becomes a free variable of the LP problem which is not restricted to non-negative values. Then, Equation (2.4) gets transformed into

$$P = \eta \, \rho \, g \left(u_1^2 - u_2^2\right) \tag{2.6}$$

The squared terms are separable functions and can be approximated by piecewise linear fashion. Without making any assumption on the convexity of the function, the piecewise linear representation is referred to as a special ordered set of type 2 (SOS2) according to:

$$y = y_1\lambda_1 + y_2\lambda_2 + y_3\lambda_3 + y_4\lambda_4 \tag{2.7}$$

$$x = x_1\lambda_1 + x_2\lambda_2 + x_3\lambda_3 + x_4\lambda_4 \tag{2.8}$$

$$\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1 \tag{2.9}$$

where x_i, y_i are pairs of the nonlinear function $y = f\left(x\right) = x^2$ for u_1, u_2 in Eq. (2.6). Mixed integer (MI) programming is required to ensure the interpolation condition that either one lambda or two adjacent ones are non-zero.

A primary research question in this project is, if linearization contributes to a better performance of an optimization under conservation of the model accuracy.

3.2.3 Logical Constraints

Operators in many water systems face logical constraints. Examples include a discontinuous spill range of a dam assuming that a ski jump spillway is not able to operate in a range $[a,b]$ (Case RS2) or a pump which can be off, or operate between minimum and maximum capacity.

A procedure for the RS2 case is presented in Williams (2013) on pages 169-172. The schematization reads

$$Q_S \geq 0 \tag{2.10}$$

$$Q_S + M\delta \leq M + a \tag{2.11}$$

$$-Q_S + M(1-\delta) \leq M - b \tag{2.12}$$

where δ is a binary indicator variables and M is a sufficiently large number we choose as $M = Q_{S,\max}$. The indicator variable makes sure that only one of the indicator variables is active implying that either Equation (2.11) or (2.12) gets activated.

Further research in this project will address the feasibility of adding such logical constraints in comparison to the alternative approaches such as the post-processing of continuous optimization results.

3.3 Representation of Uncertainty and Stochastic / Robust Optimization Setup

A suitable representation of uncertainty in the model forcing, model processes as well as objectives and constraints are an essential element of model predictive control applications. This is in particular important in application to water resources systems which uncertainties are usually much higher than in technical applications.

A common way to represent meteorological uncertainty is by probabilistic ensemble forecasts and derived hydrological products. The uncertainty in these forecasts is correlated in time. An example is the uncertain magnitude of a forested precipitation event which may impact the streamflow over many days. A suitable method to process the probabilistic forcing is by a scenario reduction of the ensemble as input for a multi-stage stochastic optimization. This method has been already implemented in the existing RTC 1.X software and successfully validated in real-world test cases.

Other uncertainty may not have less time-correlation at the time scale of interest. Examples are contingencies in a hydropower system with potential unit outages of turbines or generator. These may occur at any time step and often do not have a long lasting impact. The research question in this project is to better classify different sources of uncertainty and address the most suitable representation in the optimization problem. This will include an assessment of techniques such as security and chance constraints, (adjustable) robust optimization techniques among others.

3.4 Multi-objective Optimization

3.4.1 Overview

The management of most water systems fulfills multiple management objectives such as flood mitigation, water supply for domestic use or irrigation, environmental obligations, hydro-power generation, among others. The corresponding optimization problem is referred to as multi-objective optimization according to the formulation

$$\min_x \left[f_1(x), f_2(x), \ldots, f_i(x) \right] \tag{2.13}$$

where $f_1(x), f_2(x), \ldots, f_i(x)$ represent the multiple objectives.

An important step in the definition of the multi-objective optimization problem is the implementation of individual objective function terms and its interaction or trade-off in the overall objective function. Furthermore, we may formulate individual objectives as constraints of the optimization and vice versa. This section intends to introduce and discuss basic implementation options and addresses research within this project.

3.4.2 Weighting Method

The weighting method is a classical approach for multi-objective optimization problems and implements a weighted sum of the individual objective function terms to receive a scalar objective function f according to

$$f = w_1 f_1(x) + w_2 f_2(x) + \ldots + w_i f_i(x) \tag{2.14}$$

where $w_1, w_2, \ldots, w_i$ are weighting factors.

The approach is a logical choice, if the objective function terms represent the same quantity, f.e. if all represent operating costs and f describes the total operating costs of the system. On the other hand, the methods gets nontransparent from an operator point of view, if the objective function terms are nonlinear and represent quantities in different units. This is due to the fact, that the operators often find it difficult to correlate and quantify the importance of an objective with the weighting factor.

Furthermore, all objective function terms are traded off against each other. If one objective is much more important than the other, the corresponding implementation will require a much higher weighting factor for the more important objective. This can lead to badly scaled optimization problem in which the lower priority objective gets neglected by the optimizer due to numerical issues.

3.4.3 Goal Programming (Lexicographical Ordering)

Goal programming is an alternative approach to multi-objective optimization. Its basic version is referred to as lexicographical ordering. This means that the multi-objective optimization problem is solved stepwise starting with high-priority objectives, then moving to lower priority ones by keeping the prior function values as constraints.

For a simple example with two objective function terms, this reads

Step 1: $$\min_x f_1(x) \tag{2.15}$$

Step 2: $$\begin{aligned} &\min_x f_2(x) \\ &f_1 \leq f_{1,\min} + \text{optional relaxation} \end{aligned} \tag{2.16}$$

where f_1 is a high priority objective and f_2 is a low priority objective.

From an operator point of view, a lexicographical goal programming approach is often easier to apply than the weighting method, because of the transparent prioritization of the objectives. On contrary, it is computationally more expensive due to the solution of several optimization problems depending on the number of priority levels. Furthermore, it does not consider trade-offs between objectives.

The lexicographical goal programming can be easily combined with the weighting methods. In this case, the latter is used within a goal programming approach to summarize similar objective function terms, e.g. terms related with pumping costs. This reduces the number of sequential optimization problems and enables trade-off between objectives of the same priority level.

3.4.4 Treatment of Constraints

Another aspects in water resources optimization problems is the existence of hard constraints according to the formulation

$$g_L \leq g(x) \leq g_U \tag{2.17}$$

where the bounds g_L, g_U impose minimum and maximum bounds on the function $g(x)$. A feasible solution needs to fulfil Eq. (2.17). In contrary, the optimization problem has no solution, i.e. is infeasible, if Eq. (2.17) gets violated.

From our experience, the definition of hard constraints only on physical limits is not critical and always leads to feasible solutions. Examples include the minimum and maximum capacities of hydraulic structures such as pumps, turbines and weirs or the enforcement of mass balance in a reservoir or canal reach.

The feasibility may become an issue, if hard constraints get defined for operational limits. As an example, we refer to the maximum water level in a river reach an authority wants to maintain due to flood protection. Under the assumption of an extreme inflow and a limited storage capacity of the water system, there is probably no other option than to violate the level threshold. In this case, the definition of the level threshold as a hard constraint will result in an infeasible optimization problem without a solution.

If infeasibilities become an issue, we may i) relax the hard constraint or ii) reformulate the hard constraint as a soft constraint (Table 3.1). Both options lead us back to feasible problems, but differ in its specific implementation. Whereas the relaxed hard constraint looks for the minimum relaxation to make the optimization feasible again, the soft constraint penalizes the up- and down-crossing of bounds, for example by a quadratic least-square norm.

Table 3.1 Implementation of system constraints

soft constraint	**hard constraint with relaxation**	**hard constraint**
$\max^2\left(g\left(x\right)-g_U,0\right)$ $\min^2\left(g\left(x\right)-g_L,0\right)$	$\min_c g\left(x\right)-\varepsilon \le g_U$ $\varepsilon \ge 0$	$g_L \le g\left(x\right) \le g_U$

In both cases, the former hard constraint becomes part of the objective function and therefore it is subject to a trade-off with other objective function terms. This is often not a desirable. As an example, we refer to the minimization of operating costs of pumps in a Dutch regional water system. Pump actions can be shifted by a tactical use of storage in the water system, but water levels are not allowed to leave a pre-defined range. A definition of the level range as soft constraint may trade in a violation of this range to further decrease pumping costs. This appears unacceptable to most operators.

The potential solution to this issue to a procedure to test a constraint by a soft constraint or a hard constraint with relaxation, then fix it by a hard constraint if feasible. The approach may become part of the goal programming procedure described above.

3.5 (Re-)Design of a Conceptual and Technical Framework

3.5.1 Overview

In this section, we will discuss requirements, design principles and implementation choices of the next generation RTC-Tools 2.0 (RTC2) version.

It has been based on the experience with existing resources:
- existing version of RTC-Tools 1.X,
- Matlab + TOMLAB Optimization Toolbox,
- GAMS (base package),

and an evaluation of software packages:
- Modelica / JModelica (for generic modeling)
- CasADi as algorithmic differentiation for 1st and 2nd order derivatives and interface to NLP solvers
- SWIG for interfacing C++ to Python and Matlab,
- support of MILP and MINLP solvers via NEOS

Regarding the availability of optimization algorithms, Deltares has access to IPOPT both embedded into RTC-Tools 1.X and available via Matlab or GAMS. SNOPT and several other NLP solvers become available by the TOMLAB toolbox under Matlab.

Furthermore, problem definitions in GAMS can be solved online on the NEOS website. This will be a useful option to prototype and compare solution strategies for Mixed Integer problems. A summary of available solvers is presented in Table 3.2.

Table 3.2 Selected solvers with GAMS input on http://www.neos-server.org

	Global	LP	MILP	NLP	MINLP
BARON[1]	x				x
Bonmin[2]					x
CONOPT				x	
Couenne[2,3]	x				x
CPLEX		x (QCP)	x		
Gurobi		x (QCP, QP)	x (MIQCP, MIQP)		
IPOPT[2]				x	
KNITRO				x	x
LINDOGlobal	x				x
MOSEK		x	x		
scip	x		x		x
SNOPT				x	
XpresMP		x	x		

1 Solver allows only a limited number of nonlinear functions, e.g. exp(x), ln(x), but currently does not support other functions including trigonometric functions sin(x), cos(x), see solver documentation on http://gams.com

2 open-source solvers

3 require algebraic model description via GAMS

3.5.2 Requirements

RTC2 aims to provide infrastructure to facilitate a wide range of applications, ranging from variational data assimilation, short-term management of reservoir systems, canal networks and urban water systems to water allocation studies and long-term policy analysis. To this end, a highly flexible software architecture is required. In particular, the software will need to:

- **accommodate any 0- or 1-dimensional model**, including a core library of flow models and application-specific custom elements. Higher-dimensional models may be considered for RTC3;
- **facilitate the integration within a higher-level logic**, such as multi-objective optimization using goal programming, rule-based water allocation, or any other application-specific logic;
- read and write parameters and time series from and to **standardized file formats**.

The first two requirements open up possibilities for interdisciplinary studies, for instance by taking an integral approach to questions surrounding water, energy, and food. At the same time, it is recognized that RTC2 should be able to **run existing configurations with little or no modification**. Furthermore, remove redundancies in the existing configuration.

The software architecture must accommodate mathematical re-interpretations of an optimization problem. A modeler may, for instance, want to assign uncertainty to certain inputs, parameters or the model structure. RTC2 must be able to **assign ranges or discrete sets to parameters** (leading to robust optimization), or **ensembles of input time series** (leading to stochastic optimization). To facilitate this, an **internal symbolic representation of the optimization problem** is required. Similarly, the internal symbolic representation will allow for various techniques of **mixed-integer optimization**, including automated derivation of convex relaxations and possibly even techniques of interval analysis for a general, non-convex branch-and-bound method. This requirement is key to facilitate the research envisioned within the present research project.

Last but not least, i**ntegration with other software packages** must be provided for. These packages include:

- **RIBASIM**. The possibility of sharing the same computational core between RTC-Tools and RIBASIM 2.0 is being investigated. RIBASIM uses rules to allocate water from sources to end users, based on water distribution models. Ideally, RIBASIM 2.0 models should be compatible with RTC-Tools, so that RTC-Tools can apply optimization and distribution rules to them.
- **WANDA**. Model predictive control of pipeline networks as used in, e.g., district heating systems is an application being studied at Deltares. Use of WANDA pipe network models within RTC-Tools, with or without an intermediate conversion step, would be highly desirable. Along the same lines, optimization of lock schedules to minimize salt intrusion requires a strong coupling between the models available in WANDA Locks and RTC-Tools.
- **W-Flow**. Provision of upstream boundary conditions for RTC-Tools models from a W-Flow model is to remain supported through a **BMI** interface. Representation of W-Flow routing networks in RTC-Tools to make integrated models of reservoir systems and downstream routing reaches.
- **SOBEK**. A time step-based coupling of SOBEK and RTC-Tools feedback controllers is to remain supported through an **OpenMI** interface.
- **Delft3D Flexible Mesh**. A time step-based coupling of Delft3D Flexible Mesh and RTC-Tools feedback controllers is to remain supported through a **BMI** interface.
- **Delta Shell.** Using RTC-Tools 1.x, only feedback controllers can be set up inside Delta Shell. For the new architecture, we also want to open up the possibility of **creating model schematizations from within Delta Shell**.
- **Delft-FEWS**. It is to remain transparent to use RTC-Tools as an adapter from FEWS. A FEWS time series format (PI XML or NetCDF CF[2]) is to remain supported by the RTC-Tools core.
- **Third party optimization solvers**. RTC-Tools 1.x provides a **MATLAB** interface to the optimization problems generated by RTC-Tools, allowing them to be interfaced with alternative optimization packages through, e.g., TOMLAB. This interfacing capability is to be maintained in RTC2.
- **Python**. RTC2 is to be extensible and scriptable in Python, in line with other Deltares software such as Delta Shell and W-Flow.

For completeness, the most important technical requirements are presented below:

- **Time step independence**. By changing the time step, the rest of the configuration should not need to be changed.

---

[2] *CF (Climate and Forecast) - http://cfconventions.org/*

- **Simulation and optimization mode** including a co-simulation option via OpenMI and BMI interfaces
- **Decouple the simulation and optimization time steps** and enable dedicated aggregations (equidistant, non-equidistant, tree-based) for optimization variables
- Enable **decoupling of optimization horizon** (e.g. daily) from full time series horizon (e.g. month)
- Support for **single, multiple, and hybrid shooting** optimization methods.
- Support for **ensemble execution** with support for scenario trees and variations of the model parameters and structure in the ensemble members
- Support for smooth lookup tables using, e.g., **B-splines.**
- Computation of **first and second order derivatives.**
- Internal use of **Sparse matrices** for the computation of derivatives**.**
- **Parallel execution** whenever possible**.**
- **Performance equivalent or better than that of RTC-Tools 1.x.**

Fine-grained requirements were also derived and documented in a model-based systems engineering tool. Discussion of all fine-grained requirements, many of which are mathematical in nature, is beyond the scope of this document. Interested readers may contact the authors for access to the full set of requirements.

3.5.3 Design principles

The requirements of the previous section are more easily facilitated if a set of more general design principles are formulated. Five design principles are proposed in the present section.

- **Independent innovation is facilitated** in modeling, in the formulation of optimization problems, and in the embedding of optimization problems in higher level logic. Valuable, re-usable innovations are integrated back into the main code base.
- The **physical models are separated from their temporal discretization** whenever possible. In other words, temporal integration is, generally speaking, the domain of RTC-Tools, and not of the modeler. Spatial discretization is explicit in the modeler’s choice of nodes and branches, like it is in packages such as SOBEK.
- **Existing tools are used as much as possible** to reduce the workload of developers and modelers alike. These tools may be modeling tools like Delta Shell.

- The scope of RTC2 is limited to **zero- and one-dimensional systems**. 2D and 3D models are left for future developments towards RTC3, where they will become relevant for applications such as optimal dredging or water quality management.
- The use of high-level programming languages is preferred to enable **agile development**. Core components of the software are implemented in C/C++ and equally available in Python and Matlab through SWIG to enable rapid prototyping and custom applications. Core components are optionally implemented in Python before migrating them to C/C++.
- **Easy deployment** as an operational system. The software should come bundled with all third party dependencies.

•

3.5.4 Implementation choices

3.5.4.1 Modeling

RTC2 aims to support a much larger class of applications than RTC1. In addition to providing a fixed model library that can be changed or extended only in code, as RTC1 does, the aim is now to allow users to include any custom models that they may need. To achieve this, RTC2 will need to allow the specification of models in a generic modeling language.

From the model description, it will, as mentioned in the requirements, be necessary to be able to generate a symbolic representation. Using the symbolic representation, it will be possible to compute derivative information. Derivative computation will be discussed in the next section.

As a proven technology, fitting our requirements for a generic, user-friendly modeling process and compilation to a symbolic representation, Modelica is a natural choice for RTC2.

In order to make the modeling process as user-friendly as possible, RTC2 may be bundled with a Modelica library of hydraulic and hydrodynamic standard components. A Modelica graphical user interface, such as Wolfram SystemModeler or OpenModelica, may then be used to compose a full system model using drag and drop.

Furthermore, to facilitate GIS integration, a plug-in may be developed for Delta Shell that exposes the components contained in a Modelica package as a network schematization toolbar. In this way, geospatial Modelica models could be built directly on top of a map in Delta Shell. By developing such a Delta Shell plugin in a generic way, modelers could make their own, new Modelica components available instantaneously within Delta Shell as well, streamlining the innovation process in- and outside of Deltares.

Yet we also want to support existing RTC1 models in the new framework. To this end, the RTC1 model library and configuration parser will be part of the RTC2 framework.

3.5.4.2 Derivative computation

The manual coding of the first order derivatives in RTC1 is a process prone to error. This has previously resulted in some bugs that were hard to find. Therefore, for RTC2, it was decided to make use of *algorithmic differentiation*. Algorithmic differentiation is a family of techniques to automate the computation of derivatives of pieces of code.

RTC2 will make use of CasADi to compute first and second order derivatives. A parallelization of the CasADi function execution is intended to be implemented in the future.

3.5.4.3 Optimization

RTC-Tools needs a mechanism to define the objective function and constraints. In RTC-Tools 1.X, an XML file was used that allowed the modeler to compose a number of weighted objective terms and constraints using pre-defined elements.

For RTC2, the aim is to provide the modeler with the freedom to innovate. To this end, a generic mechanism for specifying the objective function and constraints is needed. Several options exist:

1. An unofficial extension to the Modelica language, called *Optimica*, allows one to formulate optimal control problems.
2. Dedicated Modelica classes could be used to specify objectives and constraints.
3. The constraints and objectives could be written as CasADi symbolics directly.

Integrating an unofficial language extension into an unaffiliated product is not desirable. This leaves options two and three, both of which have their attractions. The use of Modelica classes suits well to the choice of Modelica as modeling language. On the other hand, a common use case is to script the generation of objectives and constraints. A typical example is the implementation of a goal programming approach to multi-objective optimization, where the list of objectives and constraints changes at every iteration of the goal programming algorithm.

Without a clear-cut advantage of 2 over 3 or the other way around, it is suggested to support both options and backwards compatibility to the existing XML layer for the definition of objectives and constraints in RTC1.

Due to the generic nature of Modelica models, an optimization solver that is able to solve non-linear optimization problems is required. The open source interior point optimizer *IPOPT* fulfills this requirement. IPOPT is already in use as the optimizer of choice in RTC-Tools 1.x.

At the same time, it is planned to keep the optimizer interface generic, so that alternative solvers may be interfaced as required by the application through Python / CasADi or Matlab.

3.5.4.4 Data formats

RTC-Tools is commonly used in conjunction with Delft-FEWS. Consequently, support for Published Interface (PI) XML files, or alternatively for the Climate & Forecast NetCDF format, is required. RTC-Tools 1.x supports PI XML files. Delft-FEWS, on the other hand, is steadily moving towards NetCDF.

For compatibility reasons, it is suggested to keep supporting PI XML. NetCDF support will be added in the future.

Furthermore, support for other file formats may be implemented on a case-by-case basis. For instance, for the Quick Scan Tool under development for Rijkswaterstaat, CSV data I/O with Excel is required. The software framework must be set up such that new file formats can be implemented without recompilation of the core.

3.5.4.5 Ensembles

RTC-Tools is to continue to support the importing, processing and export of ensembles of model pools and predictions. It must continue to support the generation of scenario trees to formulate efficient stochastic optimization set-ups.

The implementation should continue to support parallelization options in the function evaluation and the computation of derivatives.

3.5.4.6 Programming languages

Every control or decision support system has its peculiarities. This calls for an extensible software framework, where highly application-specific functionality can be developed without burdening the RTC-Tools core.

At Deltares and in the Dutch water sector in general (e.g., Nelen & Schuurmans and Deltares' W-Flow) the scripting language Python enjoys increasing popularity. A large number of engineers and hydrologists know how to use Python at least to some extent. This stands in contrast to the classical programming language C++, knowledge of which is generally reserved to a select few.

Another scripting language that is widely known and popular for prototyping is MATLAB. MATLAB, however, is a commercial package, which makes deployment of scripts written in its language cumbersome. Furthermore, Python's object system is better designed and consequently easier to use.

There are two options to enable extensibility using Python. One is to develop RTC2 itself in Python, using Cython-precompilation, or embedded components written in C/C++, to guarantee a level of computational performance equal to that of software written in pure C. This has the advantage of introducing the compactness, readability, and agility of Python code to the entire RTC software stack.

The alternative is to write the core in C or C++, and to expose the various C++ classes through a SWIG Python / Matlab layer. Relative to a pure Python implementation, a C++ implementation requires more code, and requires explicit memory management. On the other hand, using C/C++, code can be hand-optimized to be more efficient. Furthermore, using the C++11 standard, much of the traditional boilerplate code prevalent in C++ code may be avoided.

RTC2 aims to be malleable and adaptable, so that it can be used and extended for years to come. C++11 in combination with a SWIG Python / Matlab layer is therefore a natural choice.

With Python, however, deployment is not always trivial. Python 3 scripts often don't work with Python 2 interpreters and the other way around. One should therefore bundle the appropriate Python interpreter version, together with the required packages, with one's Python scripts. There are several ways to do this, one of the cleanest ways being to use the *virtualenv* package. Using virtualenv, one creates a directory, in which the application lives, together with the interpreter executable, the Python dependencies, and any additional DLLs the application may need. This directory can be set up to be completely independent from the system-wide Python installation.

Alternatively, a full container system, such as Docker, may be used to bundle the application with all its dependencies.

The use of a cross-platform compiler such as Clang will be evaluated and used.

3.5.4.7 *Deployment*

When deploying a piece of software, such as RTC-Tools, one often runs into subtle incompatibilities between computers. A client's computer system may, for instance, have a slightly older Windows version than the one used at Deltares. Such discrepancies can, and generally speaking will, cause small changes in behavior than can be very hard to trace.

Static linking may be considered to reduce the number of dependencies to shared libraries.

In recent years, and in recognition of this problem, so-called *containers* have become popular. A container is, conceptually speaking, a minimal filesystem containing a piece of software, and particular versions of the software's dependencies. The container is then typically run on the host OS, sharing its kernel, but operating solely (*sandboxed*) within the container's own filesystem. Regardless of the underlying hardware and operating system version, the software will then use exactly the same versions of its dependencies. This minimizes the chance of library incompatibilities, and reduces the burden of the troubleshooting process.

Containers are also used to ease deployment of software in the cloud. A container can be cloned any number of times, and so may be used to scale the number of running processes on the fly. While out of scope for the present research project, a cloud-based decision support service will likely be of interest in the long term.

The most widely used container technology at this point in time is Docker. Docker originates from Google, where over 2 billion containers (The Register, *EVERYTHING at Google runs in a container*, May 2014) are launched every week. It has since also been adopted by Microsoft, which plans to launch native Docker support in Windows Server 2016 (Docker Blog, August 2015), a technical preview of which is already available. On older versions of Windows, Docker makes use of a Linux virtual machine under the hood. In this case, the use of a virtual machine does cause a performance penalty.

In RTC-Tools, one can distinguish two tasks. The first is to set up a model and an optimization problem, and to generate a symbolic CasADi function representing the entire, discretized optimization problem. The second task is to take this symbolic function, to interface it with the optimizer, and to let the optimizer locate an optimum solution.

The first task involves either the RTC1 library, or the JModelica.org compiler. Especially the latter comes with a string of dependencies. To ease installation, Docker may be a good choice.

The second task is the one that runs live in an operational system. The use of a Linux virtual machine on older Windows versions may be undesirable in an operational context. In this case, Docker is not the best choice. Instead, a Python virtualenv environment may be used. Or, alternatively, the second task may be so limited in scope and easy to isolate that a pure C++ implementation would not limit innovation. In that case, native executables could be provided for older systems, while a Docker container is delivered to customers running Windows Server 2016 or newer.

For the time being we will support both options and reassess its use in the future.

3.5.5 Control flow

This section illustrates show how a control flow may be synthesized from the considerations discussed in the preceding sections.

Figure 3.1 shows how a dedicated shared library containing the entire discretized optimization problem may be generated either from RTC1 components, or from a Modelica model.

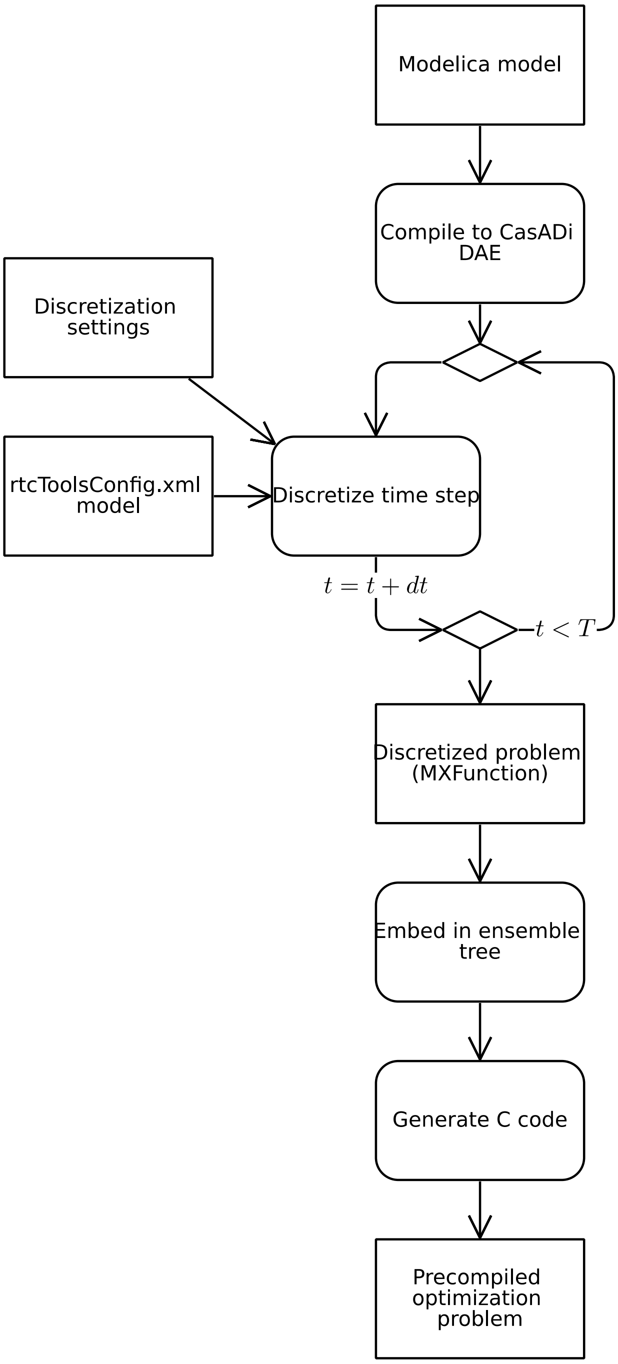


Figure 3.1 Compilation of the model and optimization problem to a dedicated shared library.

Figure 3.2 shows how the resulting symbolic function can have its parametric and time-dependent unknowns filled in from FEWS, before being interfaced with the optimization algorithm.

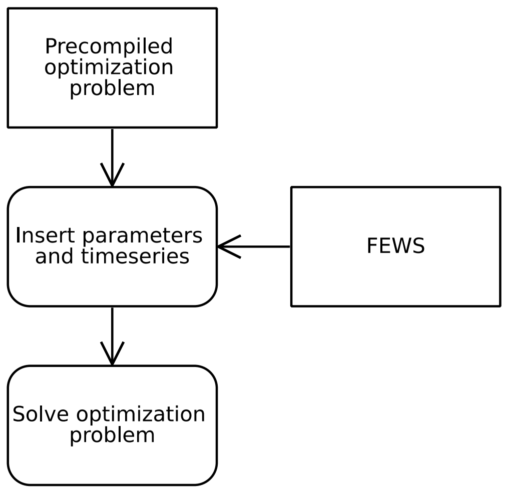


Figure 3.2 The provision of data to and optimization of the precompiled optimization problem.

4 Statement of Work

4.1 Overview

The project is executed in the period July 2015 – June 2018. The first 6 months are dedicated to an inception phase in Work Package 1. It was executed in the period July – December 2015 and is documented in this report. Work Package 2 starts at the end of the inception phase (WP1) and includes the design, implementation and feasibility assessment of novel optimization techniques. Work package 3 starts in July 2016 and covers the exploitation and dissemination of results.

The work packages are described in the following sections. Furthermore, we summarize the deliverables of the project and the related time schedule.

4.2 Work Package 1 – Inception Phase (Month 1-6)

Activities in WP1 include:

1. Inventory of the need for decision support in the flood management domain in collaboration with several stakeholders and the definition of a number of representative academic test cases
2. Inventory of state-of-the-art approaches in the field of stochastic and robust optimization and the representation of mixed-logical systems as well as the design of a new conceptual framework to support water managers.

Results of the inception phase are documented in this report.

4.3 Work Package 2 – Design, Implementation and Feasibility Assessment

4.3.1 Design Phase (Month 7-9)

<u>Task 2.1: Design Phase</u>
Resources: Deltares (Baayen, Schwanenberg), N&S

The objective of the design phase is the definition of a next generation software architecture of RTC-Tools, referred to as RTC2, based on the requirement of the inception phase. This architecture will be the basis for further research and development effort within this project.

The design phase started January 2016 and has been finalized until the end of March 2016. This report provides a high-level summary of the results. Further details on the architecture have been documented in the product management tool of RTC-Tools and become available to interested readers on request.

4.3.2 Implementation and Assessment Phase (Month 10-24)

The implementation and assessment phase includes the following activities:

<u>Task 2.2: Software Architecture RTC2 (together with RTC-Tools PM)</u>

Activities:
- CasADi Integration RTC1 (Schwanenberg, Baayen)
  Enhancement of RTC1 regarding by introduction of algorithmic differentiation by CasADi, support of 2^{nd} order derivatives, removal of redundancies in configuration related to constraints
- Python Prototyping RTC2 & Modelica Library (Baayen, den Toom)
  RTC2 prototype in Python with focus the tool chain Modelica/JModelica/CasADi, Modelica library of the TKI test cases
- Modelica Model Library (RTC-Tools PM)
  The extension of the Modelica Model Library is continued under the RTC-Tools PM project
- Cases: Academic Tests (Deltares, N&S)
  Assessment of RTC1 / RTC2 by implementation and assessment of selected academic test cases of chapter 2 starting with the application of the hydraulic model.
- API Definition RTC2 (cooperation RTC-Tools PM)
  Design of the new C++/SWIG/Python/Matlab software architecture
- Implementation of Release Candidate RTC2 (cooperation RTC-Tools PM)

<u>Task 2.3: Multi-objective Optimization</u>

- Implementation of several multi-objective optimization approaches in application to the academic test case defined in chapter 2
- Assessment of the pros and cons of every technique in terms of performance and transparency for the user

<u>Task 2.4: Hybrid System (in cooperation with WEX project)</u>

This task is executed in close cooperation with the WEX project.

Activities include:
- Implementation of several techniques to linearize nonlinear systems, model hybrid system and consider logical constraints
- Assessment of the pros and cons of these techniques in application to regional Dutch water systems

Task 2.5: Stochastic / Robust Optimization

- Classification and quantification of sources of uncertainty in Dutch water systems, i.e. large-scale, highly time correlated meteorological uncertainty, uncertainty and risk resulting from contingencies and outages, etc.
- Implementation of several techniques to represent these uncertainties in optimization problems to generate robust management decisions, we will address multi-stage stochastic optimization, (adjustable) robust optimization, the use of security and chance constraints etc.
- Assessment of the performance of these methods in application to the academic test cases and outlook to its application for real-world cases

4.3.3 Selection and Refinement Phase (Month 25-36)

Task 2.6: Selection and Refinement

The last year is dedicated to the conceptual and technical refinement of a selected approach of Tasks 2.3 – 2.5. It is intended to apply these to more integrated and larger-scale problem setups to increase its maturity level and validate its applicability to real-world problems.

Results of this task will be published in a manual “Best Modeling Practice with RTC-Tools”.

4.4 Work Package 3 – Exploitation and Dissemination (Month 13-36)

Task 3.1: National Exploitation and Dissemination (N&S)

The national exploitation and dissemination of results will be closely coordinated with the TKI project “Slim water management” and the WEX project. Both are applied research project and include a number of pilot applications to regional Dutch water systems. Both are downstream users of the techniques developed in this project.

Further exploitation and dissemination activities will be conducted by N&S. This includes the presentation of results at national conferences, workshop or software days (national software days Delft-FEWS). This includes the organization of two dedicated events to present results of this project to stakeholders.

Results will be published in Dutch journals such as H2O or comparable.

Task 3.2: International Exploitation and Dissemination

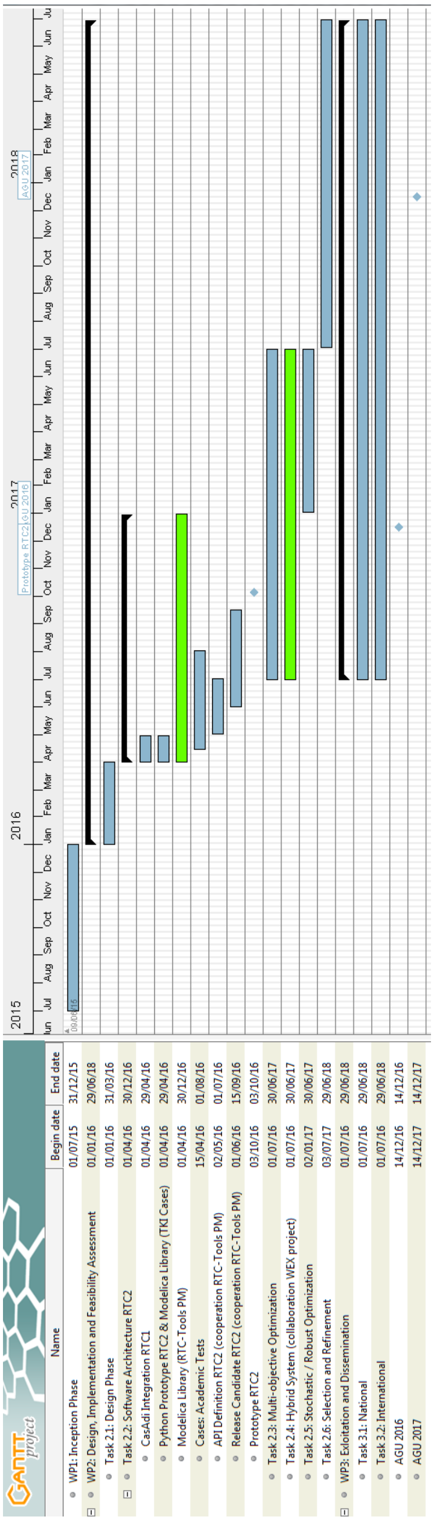
The international exploitation and dissemination is led by Deltares.

Scientific results will be published in international, peer-reviewed journals. We expect at least 3 publications. Furthermore, results will be presented at scientific conferences such as AGU, EGU or the HEPEX meetings as well as at software days (international software days: Delft-FEWS Users Meeting).

4.5 Deliverables

- Prototype of the new RTC2 software architecture in October 1, 2016 (working prototype and documentation)
  In the following, new software features will be released and documented together with the standard RTC2 release schedule
- 2 national publications
- 3 international publications in peer-reviewed journals related to the results in Tasks 2.3-2.5
- Best Practice manual as a result of Task 2.6

4.6 Time Schedule



5 References

Alvarado-Montero R, Schwanenberg D, Krahe P, Helmke P (2016): Adjoint Modeling of Hydrological Flow Routing for Variational Data Assimilation and Model Predictive Control (submitted to J. Hydrology)

Andersson, J., Gillis, J., Diehl, M. (2015): User Documentation for CasADi v2.3.0+283.4b6bc45, August 18, 2015

Griewank, A., Walther, A. (2008): Evaluating Derivatives, Principles and Techniques of Algorithmic Differentiation, SIAM

Horebeek, T. van (2015): Intelligente sturing ter beperking van overstromingen in het Demerbekken aan de hand van RTC-Tools, MSc thesis, KU Leuven (in Dutch)

Lindstrom G, Johannson B, Persson M, Gardelin M, Bergstrom S, Development and test of the distributed HBV-96 hydrological model, Journal of Hydrology 201, 272–288

Schwanenberg D, Alvarado-Montero R (2016): Total Variation Diminishing and Mass Conservative Implementation of Hydrological Flow Routing (submitted to J. Hydrology)

Williams, H.P. (2013): Model Building in Mathematical Programming, 5th edition, Wiley

Wolfs, V., Meert, P., & Willems, P. (2015). Modular conceptual modeling approach and developer for river hydraulic simulations. Environmental Modelling & Software 71 (2015) 60-77

A Appendix

A.1 Derivatives for NLP

This section summarizes the required derivative information of several optimizers (IPOPT, SNOPT, CONOPT, KNITRO) to solve an optimization problem according to

$$\min_x f(x) \tag{2.18}$$

$$g_i(x) \le 0 \tag{2.19}$$

The summary of user-supplied derivative information for several state-of-the-art optimizers is provided in Table 5.1.

Table 5.1 User-supplied derivative information for several state-of-the-art NLP approaches

	1st order derivatives	**2nd order derivatives**
IPOPT	objective function gradient $\nabla f(x)$ constraint Jacobian $\nabla g(x)^T$	option "limited-memory" - option "exact" (Hessian of Lagrangian) $\sigma_f \nabla^2 f(x) + \sum_i \lambda_i \nabla^2 g_i(x)$ where σ_f, λ_i are provided
SNOPT	see above	-
CONOPT	see above	Hessian of Lagrangian $\sigma_f \nabla^2 f(x) + \sum_i \lambda_i \nabla^2 g_i(x)$ directional derivative* (Hessian vector product) $\sigma_f \nabla^2 f(x) u + \sum_i \lambda_i \nabla^2 g_i(x) u$ where u is the direction
KNITRO	see above	either Hessian of Lagrangian or Hessian vector product, see above

* directional second derivatives are used when the expected number of iterations in the SQP sub-solver is low and the Hessian is used when the expected number of iterations is large

Table 5.2 provides an overview about existing constraints and objectives in RTC1 and addresses its contribution to the 2nd order Langrangian Hessian matrix.

Table 5.2

constraints and objectives in RTC-Tools	**treatment in (IPOPT) optimizer**	**contribution to Lagrangian Hessian (to be implemented)**
<u>constraints</u>		
bounds on optimization variables: $x_{\min} \le x^k \le x_{\max}$	dedicated definition as variable bounds	0
rate-of-change constraints on optimization variables: $\Delta x_{\min} \le x^k - x^{k-1} \le \Delta x_{\max}$	(linear) constraint	0
average constraint on optimization variable: $\overline{x}_{\min} \le \frac{\sum_{i=k-1}^{k} x^k}{n+1} \le \overline{x}_{\max}$	(linear) constraint	0
bounds on states or model outputs: $y_{\min} \le y^k\left(x\right) \le y_{\max}$	(nonlinear) constraint	non-zero depending on the process model
<u>objective function terms</u>		
penalty of setpoint deviation: $\left(x^k - x_{sp}\right)^p$	no distinction between terms related to optimization variables, model states or model outputs, treatment as nonlinear objective function term	no contribution for linear term, constant contribution in combination with non-linear process model for quadratic terms
rate-of-change penalty of setpoint deviation: $\left(x^k - x^{k-1} - \Delta x_{sp}\right)^p$		
average penalty of setpoint deviation: $\left(\frac{\sum_{i=k-n}^{k} x^i}{n+1} - x_{sp}\right)^p$		reformulation for large n by introduction of additional optimization variable z^k to avoid a dense Hessian: $z^k = \frac{\sum_{i=k-n}^{k} x^i}{n+1}$ (linear constraint) $\left(z - x_{sp}\right)^p$

A.2 Algorithmic Differentiation Options for 1st and 2nd Order Derivatives

This sections presents the forward-over-reverse algorithmic differentiation to compute 2nd order derivatives.

The input x and the tangent $\dot{x}$ for the derivative computation are given. The variables $\bar{x}$ and $\dot{\bar{x}}$ denote the first-order and second order adjoint vectors, respectively. The functions $F(x)$ represents the dependency of the model output y on x. Then, the 2nd order derivative is computed stepwise with the procedure below:

	first-order derivative	second-order derivative
step 1: forward sweep	$y = F(x)$	$\dot{y} = F'(x)\dot{x}$
step 2: reverse sweep	$\bar{x}^T += \bar{y}^T F'(x)$	$\dot{\bar{x}}^T += \dot{\bar{y}}^T F'(x) + \bar{y}^T F''(x)\dot{x}$

Initialization is required for the tangent or search direction $\dot{x}$ and the adjoint vector $\bar{y}$.

As an example, the procedure is applied to the scalar-valued function:

$$y = F(x) = \sin(x_1)\,x_2$$
$$F'(x) = \begin{bmatrix} \cos(x_1)\,x_2 \\ \sin(x_1) \end{bmatrix},\; F''(x) = \begin{bmatrix} -\sin(x_1)\,x_2 & \cos(x_1) \\ \cos(x_1) & 0 \end{bmatrix} \tag{2.20}$$

where F', F'' are the 1st and 2nd order derivatives of y with respect to x. Then, the forward-over–reverse mode becomes

	first-order derivative	second-order derivative
step 1: forward sweep	$y = \sin(x_1)\,x_2$	$\dot{y} = \cos(x_1)\,x_2\,\dot{x}_1 + \sin(x_1)\,\dot{x}_2$
step 2: reverse sweep	$\bar{x}_1 += \bar{y}\cos(x_1)\,x_2$ $\bar{x}_2 += \bar{y}\sin(x_1)$	$\dot{\bar{x}}_1 += \dot{\bar{y}}\cos(x_1)\,x_2$ $+\bar{y}\left[-\sin(x_1)\,x_2\dot{x}_1 + \cos(x_1)\,\dot{x}_2\right]$ $\dot{\bar{x}}_2 += \dot{\bar{y}}\sin(x_1) + \bar{y}\cos(x_1)\,\dot{x}_1$

If y is an objective function and we want to compute its gradient and Hessian for $x = (x_1, x_2)^T$, we initialize $\bar{y} = 1$, $\dot{\bar{y}} = 0$ and execute the procedure above with $\dot{x} = (1,0)^T$ and $\dot{x} = (0,1)^T$ to receive the first and second column of the Hessian, respectively.

In case of a directional Hessian, the procedure is executed once with the related search direction or tangent. If a full Hessian is required for larger problems with n dimensions, graph coloring techniques can significantly reduce the computational effort for a sparse Hessian by less than n execution of the procedure above.

A.3 Algorithmic Differentiation of Implicit Functions

In case of an implicit function, the computation of derivatives relies on the application of the implicit function theorem in combination with the adjoint sensitivity equation. We present a simplified procedure from Griewank & Walther (2008), pp. 370-373 with $y = f\left(z, x\right) = z$ in application to a simple implicit reservoir model (fixed crested spillway with the uncontrolled spillage $Q_S\left(FB\right) = FB^{2/3}$ and a controlled turbine flow Q_T).

The index * denotes a solution at a root or sufficiently close to it. Enhanced methods and derivative quality criteria are presented in Griewank & Walther (2008).

	general formulation	example
independent variables	x	$x = \left[Q_I^k \quad Q_O^k \quad FB^{k-1} \right]^T$
dependent variables (system states)	z	$z = \left[FB^k \right]$
system of nonlinear equations	$w \equiv F\left(z, x\right) = 0$	$F\left(z,x\right) = \left(FB^k - FB^{k-1}\right)\frac{A}{\Delta t} - Q_I^k + Q_T^k + \left(FB^k\right)^{2/3}$
	$F_x\left(z,x\right) = \frac{\partial}{\partial x} F\left(z,x\right)$ $F'\left(z,x\right) = \frac{\partial}{\partial z} F\left(z,x\right)$	$F_x\left(z,x\right) = \begin{bmatrix} -1 \\ 1 \\ -A/\Delta t \end{bmatrix}$ $F'\left(z,x\right) = \left[\frac{A}{\Delta t} + \frac{3}{2}\left(FB^k\right)^{1/2} \right]$
adjoint sensitivity equation	$0 = \bar{F}\left(z_*, x, \bar{w}_*, \bar{y}\right) \equiv F'\left(z_*, x\right)^T \bar{w}_* - \bar{z}$	$\left[\frac{A}{\Delta t} + \frac{3}{2}\left(FB^k\right)^{1/2} \right] \bar{w}_* = \bar{z}$ scalar function, requires the solution of a linear equation system for vector functions
adjoint of independent variables	$\bar{x}_*^T = -\bar{w}_*^T F_x(z,x)$	$\bar{x}_*^T = -\frac{\bar{z}}{\frac{A}{\Delta t} + \frac{3}{2}\left(FB^k\right)^{1/2}} \begin{bmatrix} -1 \\ 1 \\ A/\Delta t \end{bmatrix}$

A.4 Features of Software Packages

	RTC-Tools	**GAMS**	**CasADi**
Model Library	discrete schematization and integration, so far only 1st order derivatives (all coded by the user), Modelica integration by FMI in progress	discrete schematization by user but no integration, framework automatically generates 1st and 2nd order derivative information	discrete or continuous equations in AD formulation or via Modelica (user), framework computes 1st and 2nd order derivatives by algorithmic differentiation
Deterministic Optimization Setup	collocated, direct single and multiple shooting setup depending on the model implementation hybrid system via Matlab and Python interfaces (both need extensions) and scripts with solvers in Matlab/Python	only collocated setups native support of binary and integer variables as well as dedicated embedded solvers	collocated, direct single and multiple shooting setup independent of the model no support for hybrid systems this point (integration with C++/Python/Matlab interface ?)
Representation of Uncertainty and Stochastic Optimization Setup	scenario tree generation and tree-based optimization embedded	scenario tree generation (?) and tree-based optimization embedded	no support at this point (integration by C++/Python/Matlab interface ?)
Multi-objective optimization	scripting in Matlab and Python (will require interface extensions), embedded features in a later stage	direct GAMS scripting based on available examples of the model library	no support at this point (integration by C++/Python/Matlab interface ?)