



**HAL**  
open science

## Generating Hydro Unit-Commitment Instances

Dimitri Thomopoulos, Claudia d'Ambrosio, Wim van Ackooij, Marc Stéfanon

► **To cite this version:**

Dimitri Thomopoulos, Claudia d'Ambrosio, Wim van Ackooij, Marc Stéfanon. Generating Hydro Unit-Commitment Instances. TOP An Official Journal of the Spanish Society of Statistics and Operations Research, inPress, 10.1007/s11750-023-00660-w . hal-04179739

**HAL Id: hal-04179739**

**<https://hal.science/hal-04179739v1>**

Submitted on 10 Aug 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Generating Hydro Unit-Commitment Instances

Dimitri Thomopoulos      Wim van Ackooij  
Claudia D'Ambrosio      Marc Stefanon

Received: date / Accepted: date

## Abstract

Handling cascading reservoir systems is an important energy management optimization problem. The difficulty of these problems stems, in part, from the modeling of the hydro-production function. Data are not always easy to come by. To remedy this issue, this paper proposes and describes a realistic instance generator, building instances of varying difficulty. No specific data format for these problems is currently available in the literature. In addition, both notation and formulations tend to vary widely. Instances and case studies are poorly available or otherwise not applicable to different variants of these problems. The purpose of the generator is also a first attempt to develop a shared data format useful for different variants of hydro unit commitment problems and seeks to mediate the needs of different sectors interested in the optimization of hydro power plants. Hydro Unit Commitment Instance Generator Cascading Reservoir System

## 1 Introduction

Managing the electrical system on a day to day basis involves detailed modeling of the system as a whole. Traditionally this management is achieved through unit-commitment, e.g., [19, 24]. In energy systems involving both thermal units and cascading systems, the latter do add quite some complexity to the overall solution process. We refer to [20] for a detailed overview of cascading reservoir management. Centrally managing these resources is still vital today as carefully explained in [24]. In particular, water is a key resource for several economic activities such as energy, agriculture, and tourism.

Efficiently solving these unit-commitment models is, thus, important for practice. The specific nature of managing cascading systems immediately steps forth. First of all, the cascading system naturally connects a variety of units, whether turbines or pumps, together through the overall cascading nature of the entire system. The natural connection is through the linear flow equations involving the release rates of each turbine (and pumping rates). The issue is, however, that the flow rate of a turbine is not necessarily nicely connected to the amount of power output. These so-called hydro-production functions are

typically non-concave and a vast literature of approximations is available. We refer to [5,9–11,13,14,16,26] and the references therein for more information. In some situations, it may be convenient to actually discretize the hydro-production function, so as to consider the set of feasible flow rates to be completely discrete, e.g., [23]. There are some added advantages to this way of proceeding, and evidently some disadvantages. The main disadvantages are of course that the discrete variables naturally render the whole problem more rigid, especially if overall the flow equations and reservoir bounds are “tight”. Moreover, it may well also be that a significant number of discrete variables is required to have a reasonably precise model. In terms of advantages, first of all, the non-concave nature of the hydro-production function (possibly depending on the water head), is linearized. Furthermore, it becomes possible to integrate with ease the amount of energy generated for “spinning reserve” requirements. This latter feature would otherwise require accounting for differences of the hydro-production function, not at all easy to model and handle. Indeed, it suffices to think of the easy situation wherein the hydro-production function is concave. Then, already modeling precisely such a difference of mappings leads to a “difference of convex problem”, e.g., [25]. These non-convex, possibly non-smooth problems are much harder to solve. Another possibility, as the hydro-production function is concerned, is the direct use of non-convex optimization and modeling through polynomials. It is also possible to decompose the cascading system, including the hydro-production function into smaller pieces by employing, for instance, Lagrangian relaxation, e.g., [9,10]. For a careful comparison of these approaches, we refer to [11].

It, thus, appears that cascading reservoir management is still not a well solved problem. This stems from three main sources:

- The computational time needed to solve a single instance, which may not always be negligible. One can think of this as a highly crucial point if the cascading system is furthermore inserted into a decomposition scheme of its own w.r.t. an encompassing unit-commitment problem.
- The difficulty induced by the non-convex hydro-production function.
- The need to potentially account for uncertainty on inflows (e.g., through probabilistic constraints [27]).

Therefore, this naturally explains the existence of a wide variety of solution approaches. It is, however, not evident to carefully compare various approaches as problem instances are not widely available. It will be the purpose of this work to reduce this gap, by building an instance generator. We intend to make publicly available the generator of hydro instances of a variety of sizes and difficulties. No specific data format for these problems is currently available in the literature. Instances and case studies are poorly available or otherwise not applicable to different variants of these problems. The purpose of the generator is also a first attempt to develop a shared data format useful for different variants of hydro unit commitment (HUC) problems. It is evident that a document format that covers

every possible variant is not easily achievable. This generator is therefore a first attempt to establish a format, which already covers several uses, but still to be expanded in the future with more data types or data sources, obtaining a real library of case studies, as already happens in other problem solving communities. Validating the need for a unique instance format is the presence in the literature of some attempts to provide programs and tools that can model/simulate different variants hydro power generation problems, for example, SHOP (Short-term Hydro Optimization Program) [18], which is a modeling tool for short-term hydro operation planning. The data of the generator are inspired by a specific HUC, however, it was made with the view of belonging to a format that includes many more variants and many more features. In this regard, the generator creates instances in AMPL format which is widely used in the optimization community and in csv which is a very generic format and easily readable by many software or programming languages. The code and the data can be found here: [https://people.unipi.it/dimitri\\_thomopoulos/libraries/hig/](https://people.unipi.it/dimitri_thomopoulos/libraries/hig/).

This paper is organized as follows: in Section 2 we define in detail the HUC problem we consider. In Section 3, we describe the historical data that we use for the instance generator and, when these are not available, the models we developed to generate them in a realistic way. In Section 4 we present some models which can exploit the generated instances. Section 5 is devoted to computational results aimed at showing that our generator can provide instances of increasing difficulty and instances of the same size can be extremely easy or challenging. Finally, we draw conclusions and discuss perspectives in Section 6.

## 2 Problem Definition

The HUC variant we consider is very similar to the one tackled in [5]. In fact, as we explain in Section 3, the instances we generate are also inspired by the ones solved by Borghetti et al. [5].

We focus on the deterministic price-taker variant of the HUC. Multiple reservoirs and hydro-plants can be considered, which was not the case in [5]. In the HUC we are given a short time horizon (typically 24 hours or a week) during which, we decide how to produce or consume the power of one or more hydro-plants. To produce power, turbines are activated thanks to the water in upstream reservoirs. To consume power, pumps are activated to move water from downstream to upstream reservoirs. The aim is to maximize the profit of sold power while accounting for pumping and potential startup costs for the turbine/pump units. In some periods of the time horizon, when the price of power is low, it might be convenient to consume power for pumping so as to have water available to produce and sell power in later periods in which the power cost is higher. This explains the economic interest of turbinning. In the following, the components of the problem are presented in a schematic mode in order to facilitate their identification within the format of the instance. Given an overview of the decision variables, we define the number of periods, reservoirs, turbines, and pumps. We define some general parameters. Then, for

each reservoir, turbine, and pump, we define their specific parameters. Finally, we present a set of auxiliary parameters that link interconnected information between different elements.

## 2.1 Main decision variables

The main decision variables are:

- the water flow passing through each of the turbine/pump units at each time step.
- the spillage, i.e., water spilled, from each of the reservoirs at each time step. The spilled water does not flow through turbines, it is just released. It may or may not appear in downstream reservoirs without generating energy.

These decision variables are closely related to the following derived quantities, occasionally modeled as auxiliary variables:

- the water volume in each of the reservoirs at each time step.
- the produced/consumed power by each of the turbine/pump units at each time step. This is a variable depending on the water flow passing through the same unit and the water volume in the upstream reservoir.

## 2.2 Set of general parameters

The part of the problem we focus on is data, as our contribution is an instance generator. For convenience of interpretation of the format of the instances, the following notation of the parameters of the model is the same one used in fact in the instances. The dimensions of the parameter sets and some general parameters are

- $T$ : number of time steps of the considered time horizon. It allows us to define the set PERIODS :=  $\{1, \dots, T\}$ .
- $\text{delta.t}$ : time step duration (in hours).
- $J$ : number of reservoirs in the considered hydro valley. It allows us to define the set RESERVOIRS :=  $\{1, \dots, J\}$ .
- $\text{rampup}$ : represents the limit of the water-flow increase between two consecutive time steps (in  $[m^3/s]$ ).
- $\text{rampdown}$ : represents the limit of the water-flow decrease between two consecutive time steps (in  $[m^3/s]$ ).
- $\text{theta.min}$ : lower bound on the amount of water released in each time period (in  $[m^3/s]$ ).

- $s_{max}$ : maximum water spillage (in  $[m^3/s]$ ).
- $N_{turbines}$ : number of turbines.
- $N_{pumps}$ : number of pumps.

### 2.3 Reservoir specific parameters

For each reservoir in RESERVOIRS, we have the following specific parameters:

- $v_{min}$ : lower bound on water volume in the reservoir (in  $[m^3]$ ).
- $v_{max}$ : upper bound on water volume in the reservoir (in  $[m^3]$ ).
- $v_0$ : initial water volume in the reservoir (in  $[m^3]$ ).
- $v_T$ : target water volume in the reservoir at the end of the time horizon (in  $[m^3]$ ).
- $pump\_activation\_via\_turbine$ : is the pump activated using the power produced by a turbine? 0 = no, 1 = yes (this influences how the cost of activating a pump is computed).

### 2.4 Turbine and pumps specific parameters

For each turbine in TURBINES, we have the following parameters.

- $qT_0$ : initial water flow (in  $[m^3/s]$ ).
- $g_0$ : initial status of the turbine. It takes the value 1 if the turbine is on in period 0, 0 otherwise.
- $scT$ : startup cost of the turbine, paid any time the turbine is turned on (in €).
- $q_{min}$ : lower bound on the water flow (in  $[m^3/s]$ ).
- $q_{max}$ : upper bound on the water flow (in  $[m^3/s]$ ).
- $wT_{init}$ : water needed to start up each turbine (in  $[m^3/s]$ ). It represents the water wasted at each startup.
- type: type of turbine, i.e., L for downstream of a lake, E for downstream of a dam, F for run-of-river, D for valve, and R for discharge.
- plantT: index of the power plant the turbine belongs to.

For each pump in PUMPS, we have the following parameters.

- $qP_0$ : initial water flow (in  $[m^3/s]$ ).
- $u_0$ : initial status of the pump. It takes the value 1 if the pump is on in period 0, 0 otherwise.

- *scP*: startup cost of the pump, paid any time the pump is turned on (in €).
- *wP\_init*: water needed to start up each pump (in  $[m^3/s]$ ). It represents the water wasted at each startup. It is relevant when a pump is turned on by a turbine.
- *eP\_init*: energy needed to start up each pump (in  $[MWh]$ ). It is relevant when a pump is turned on by consuming power.
- *plantP*: index of the power plant the pump belongs to.

For each turbine in TURBINES, given the set of pumps PUMPS we have also the following parameter.

- *t2p*: index of the pump associated with each turbine.  $t2p = -1$  if the turbine has no corresponding pump.

## 2.5 Forecasting parameters

Typical forecast data for HUC problems are prices and inflows. In this case some data come from historical series, as the purpose is not mainly on forecasts, but on presenting some real data useful for validation of models and approaches.

For each time period  $t \in \text{PERIODS}$ , we have

- *prices*: prices of electricity (in €/MWh).

We refer the reader to Section 3.1 for further details on how we generate these values.

For each time period and reservoir, we have

- *inflows*: water inflow in period  $t$ . The water is added to the water volume in the reservoir (in  $m^3/s$ ).

We refer the reader to Section 3.2 for further details on how we generate these values.

## 2.6 Power function parameters

In addition, the expression and data defining the produced power function in terms of the water flow and the water volume in the reservoir are needed. Thus, for each time step, turbine, and reservoir, we define

$$p = \phi(q, v), \tag{1}$$

where  $p$  is the amount of produced power (in MW),  $q$  is the water flow (in  $m^3/s$ ), and  $v$  is the water volume in the reservoir (in  $m^3$ ). A value for each of the coefficients of the closed form of the function  $\phi$  is provided as part of the instance data.

Alternatively, when the generator is not provided with an explicit function, we can consider a MILP approximation as mentioned in the introduction. The most general approximation methods discretize the domain of both variables  $q$  and  $v$ . We call the discretization points of the domain of  $q$  operational points. Each instance includes the following data.

For each reservoir

- $R$ : is the number of discretization points of the volume domain.

For each reservoir and each volume discretization point

- $V$ : the water volume corresponding to the given volume discretization point. Each of which is associated with a volume value in  $[v_{min}, v_{max}]$ : the first discretization point corresponds to  $v_{min}$ ; the volume associated with each of the following  $R - 1$  discretization points is computed as “volume of previous discretization point +  $(v_{max} - v_{min} / R)$ ”.

For each turbine in TURBINES, we have

- $nOPT$ : number of operational points (o.p.), i.e., values of water flow considered.

For each turbine and operational point

- $Q$ : water flow corresponding to each o.p. of the turbine (in  $[m^3/s]$ ).

For each turbine, operational point, and volume discretization point

- $P_{ir}$ : power produced by the turbine at the given o.p. and volume discretization point (in  $[MW]$ ).

For each pump in PUMPS we have

- $nOPP$ : number of o.p. for each pump, i.e., values of water flow considered.
- $P_u$ : power consumed by a pump at each o.p. (in  $[MW]$ ). This power is formalized as absorbed and therefore takes on negative values.
- $Q_u$ : water flow corresponding to each o.p. of a pump (in  $[m^3/s]$ ).

## 2.7 Model Constraints

The constraints and objective function we consider are similar to the ones presented in [5], except for the linearization of the power function. Namely,

- constraint on the target volume at the end of the horizon.
- water balance equations: water conservation within two consecutive time periods, while accounting for the cascading structure.
- lower and upper bounds on the flows in the turbine/pump units, including minimum water flow passing through the turbine.



- limit the flow variation within two consecutive periods (the so-called ramp-up/down constraints).
- constraint on the water spillage released at the startup of a turbine.
- lower bound on the amount of water released in each period.
- switch-on/switch-off rules of the turbines and pumps.
- no simultaneous turbine and pump mode.
- constraints related to the power production as a function of water flow and water volume (as a proxy for the head effect).

We refer the reader to Borghetti et al. [5] for more details on the mathematical modeling of HUC. Mathematical models including a standard linearization of (1) are provided on the aforementioned website. The modeling language AMPL [1] was used to code them.

### 3 Historical and Generated Data Description

The default instances that are generated are inspired by the ones solved in Borghetti et al. [5]. In particular, the authors considered a week (168 hours) from the months of April, June, and December of 2004. Thanks to our generator, we can consider any time horizon from 01/04/2004 to 31/12/2012, as explained next. All of the data used by the generator are endemic to the generator itself except for the price data, which come from data already available in the literature. In addition, although the data is currently limited to the years for which all values are available, the generator continues to be usable for different historical data if provided with files in the appropriate format regarding prices and inflows as explained in the following sections.

Two data formats are available, as will be explained in more detail below, a single-reservoir format and a multi-reservoir format. At present, the proposed data generator allows creating data instances with a single reservoir inherent to the Suviana power plant, data instances in a multi-reservoir format with a single reservoir inherent to the Suviana power plant, and multi-reservoir data instances with a cascading structure drawn from the Reno drainage basin (see, e.g., [8,21]). This drainage basin, as shown in Fig. 1 has 4 reservoirs and 5 power plants.

#### 3.1 Price dataset

As already mentioned the price dataset is already available in the literature, i.e., the historical data from the Italian Electricity Market that can be found (and downloaded) on the website <http://www.mercatoelettrico.org/>. The exact internet URL at least until December 2021 is [https://www.mercatoelettrico.org/en/Download/DownloadDati.aspx?val=MGP\\_Prezzi](https://www.mercatoelettrico.org/en/Download/DownloadDati.aspx?val=MGP_Prezzi). The historical data

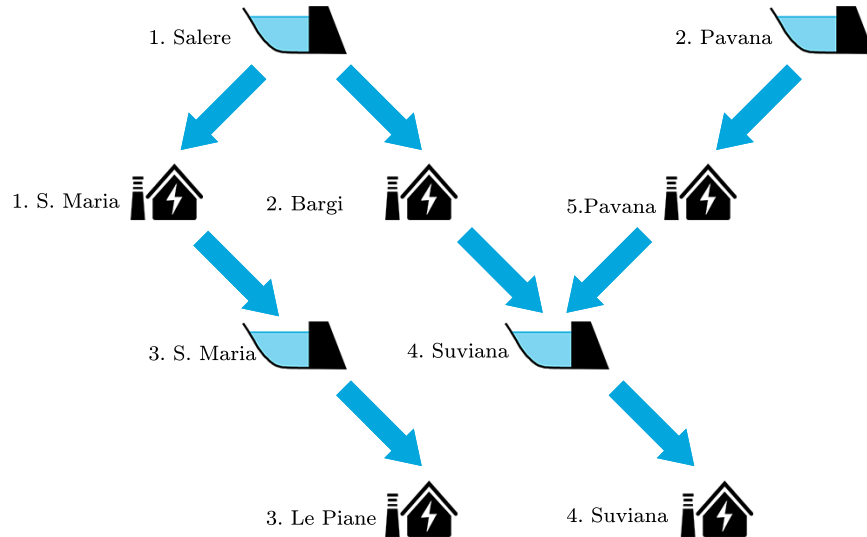


Figure 1: Reno drainage basin scheme.

starts on 1st of April 2004 and prices are national and hourly. Although usable, these data are under the copyright of the Gestore dei mercati energetici (GME), i.e., Italian Power Exchange (IPEX) which is the company responsible in Italy for the organization and management of the electricity market. Thus, we cannot provide the user with such data, thus the user should download it and use it correctly. However, to simplify this process the generator is provided with an attached script that can facilitate the search of the data, and allows to overcome technical difficulties and limitations of the site. Being dependent on third-party data for pricing, the procedure provided by the generator is designed to make the least amount of interventions to these files. In addition, even if the proposed format is using hourly resolution, the generator can also use other resolutions such as half-hour, quarter-hour, etc.

The price data format that the generator needs to generate the instances is a separate csv file for each year that the user wants to use called “Anno YYYY.csv” where YYYY is the year in question, for example, “Anno 2004.csv”. An example csv (with “;” as separator) from the year 2004 is presented in the Table 1. In the table are represented only the first 3 periods that, in the case of the Italian electricity market, correspond to one hour each. The first column includes the date in year, month, day format, i.e., YYYYMMDD. The second column indicates the reference time (between 1 and the maximum number of periods in 1 day, for example, 24 for hourly data and 48 for half-hourly data). Finally, the third column shows the national unit price (PUN) in  $\text{€}/MWh$ . The already downloadable files contain additional columns containing the market prices of individual zones and neighboring markets. However, these values are not used explicitly by the generator.

Date (YYYYMMDD)	delta.t	PUN
20040401	1	35.45
20040401	2	33.06
20040401	3	32.01

Table 1: Example of csv data of prices for the first 3 periods for the year 2004 used by the generator.

### 3.2 Inflow dataset

The historical inflow data was unavailable, besides the three weeks considered in Borghetti et al. [5]. Thus, we generated the inflow dataset with the ORCHIDEE model while using WRR2 meteorological forcing, with an hourly resolution at Suviana dam location.

ORCHIDEE (ORganizing Carbon and Hydrology In Dynamic EcosystEms) is a land surface model that simulates the fast processes occurring between the vegetated land surface and the atmosphere, such as the soil water budget and the exchanges of energy, water, and CO<sub>2</sub> through photosynthesis between the atmosphere and the biosphere [15]. It includes a complex routing scheme that routes the water to the ocean and has been validated with in-situ observations [17].

The WRR2 (Water Resources Reanalysis) forcing is based on the original ERA-Interim meteorological data bilinearly interpolated to 0.25° [7]. It includes a topographic adjustment to temperature and a recent precipitation dataset specifically designed for hydrological modeling [2].

However, even in this case, it is possible to provide the generator with data with a resolution different from the hourly, remaining consistent with the chosen resolution for prices as well.

The inflow data format that the generator needs to produce the instances is a csv file (with ";" as separator) composed of three columns as shown in an example in Table 2, i.e., date, delta.t, and inflow, respectively. The date is expressed as year, month, day, i.e., YYYYMMDD. The second column indicates the reference time (between 1 and the maximum number of periods in 1 day, for example, 24 for hourly data and 48 for half-hourly data). Finally, the third column shows the inflows expressed in  $[m^3/s]$ . Additional rows can be added to the existing file with the condition of entering the data in ascending date order. Data continuity is not required.

Figure 2 shows some examples of inflows comparing (a) the hourly inflows in the first 7 days of January 2004, (b) the daily inflows of the months of April, August, and December 2004, and, finally, (c) the daily inflows of the years 2004, 2008, and 2012.

Date (YYYYMMDD)	delta.t	Inflow
20040401	1	2.48
20040401	2	2.31
20040401	3	2.17

Table 2: Example of csv data of inflows for the first 3 periods for year 2004 used by the generator.

### 3.3 Instance formats and file extensions

Two formats of instances are available, namely single-reservoir format and multi-reservoir format. The instances in single-reservoir format omit information about the reservoirs because it is only one, while in the multi-reservoir case the parameters described in Section 2 are expressed through the set RESERVOIRS. In the last case, additional information concerning the structure of the cascading system is provided. In particular, for each turbine in TURBINES, the following parameters are provided

- $tUp$ : index of the upstream reservoir.
- $tDw$ : index of the downstream reservoir.

Both values are = -1 if the turbine has no corresponding reservoir.

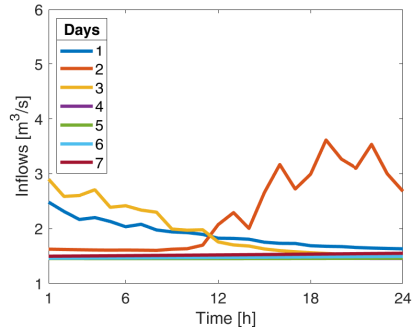
In addition, a parameter representing the delay from the upstream to the downstream reservoir is presented

- $tDelay$ : is the delay [s] in flow from the upstream reservoir to the corresponding downstream reservoir.  $tDelay = 0$  if 0 or there is not a corresponding reservoir.

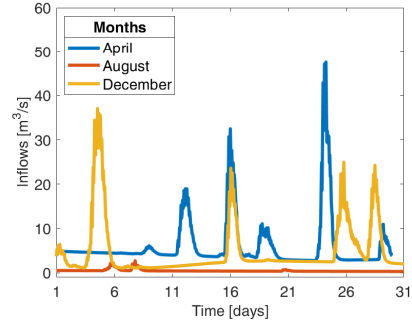
All the formats are available in two file extensions, i.e., a dat file which is typical AMPL data file format and a csv file. The two extensions were chosen because they are widely used, easily interpreted, and readable. The csv file maintains the same structure as the AMPL file, however, it has a first column where the parameters of the row are defined separated by white spaces, followed by the values of the parameters themselves. In the case of data matrices, only the first row presents a first column defining the data, while the following ones present an empty first column. The data format is the same as that used to present them in Section 2.

An example of AMPL extension data of inflows and prices for a case with 3 periods and a single reservoir is

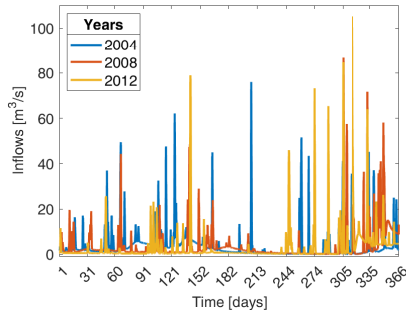
```
# T is the number of time steps considered
param T := 3;
# inflows{PERIODS} is the external (given) water inflow for each time
step [m3/s]
# prices{PERIODS} is the unit price of the power generated/consumed
at each time step [euros/MWh]
```



(a) Comparison of inflows [ $m^3/s$ ] among the first 7 days of January of the year 2004.



(b) Comparison of inflows [ $m^3/s$ ] among 3 representative of the year 2004, i.e., April, August, December.



(c) Comparison of inflows [ $m^3/s$ ] among the years 2004, 2008, and 2012.

Figure 2: Examples of historical inflows estimation.

```
param: PERIODS: inflows prices :=
1 2.48 35.45
2 2.31 33.06
3 2.17 32.01
;
```

```
While the corresponding csv format is
T;3;;;;;;;;;
PERIODS inflows prices;1;2.48;35.45;;;;;;;;;
;2;2.31;33.06;;;;;;;;;
;3;2.179;32.01;;;;;;;;;
```

also shown in Table 3.

The subsequent data are  $\Delta t$ ,  $rampup$ ,  $rampdown$ ,  $v_{min}$ ,  $v_{max}$ ,  $v_0$ ,  $v_T$ ,  $N_{turbines}$ ,  $N_{pumps}$ ,  $pump\_activation\_via\_turbine$ ,  $\theta_{min}$ , and  $s_{max}$ , respectively and each on a new line. These are followed by the turbine data

T	3		
PERIODS inflows prices	1	2.48	35.45
	2	2.31	33.06
	3	2.17	32.01

Table 3: Example of csv data of prices [ $\text{€}/MWh$ ] and inflows [ $m^3/s$ ] for 3 periods.

presented in Section 2.4 in exactly the same order, all in the same row, and with one row for each turbine. Followed again by the pump data also in the order presented in Section 2.4 and with one row for each pump. Then, followed by the number of intervals  $R$ , water flow o.p.s of turbines  $Q_i$ , power o.p.s of turbines  $P_{ir}$ , water flow o.p.s of pumps  $Q_u$ , power o.p.s of pumps  $P_u$ , volume  $V$ , and  $t2p$ . An example with  $R = 2$ , 1 reservoir, 1 turbine, 1 pump, 3 o.p.s of turbine and 2 o.p.s of pumping is

```

param delta_t := 1;
param rampup := 70;
param rampdwn := 70; param v_min := 15000000;
param v_max := 33000000;
param v_0 := 21080000;
param v_T := 21080000;
param N_turbines := 1;
param N_pumps := 1;
param pump_activation_via_turbine := 0;
param theta_min := 0;
param s_max := 0;
param: TURBINES: qT_0 g_0 scT nOPT q_min q_max wT_init type plantT:=
1 0.00 0 75.00 3 8.40 42.00 0.00 L 1
;
param: PUMPS: qP_0 u_0 scP nOPP wP_init eP_init plantP:=
1 0.00 0 75.00 2 0.00 0.00 1
;
param R := 2;
param: Q_i :=
1 1 0.00
1 2 8.40
1 3 42.00
;
param: P_ir :=
1 1 1 0.000000
1 2 1 2.816118
1 3 1 23.272352
1 1 2 0.000000
1 2 2 3.065391
1 3 2 25.906705

```

```

;
param:  Q_u P_u :=
1 1 0.00 0.00
1 2 -26.98 -21.40
;
param:  V :=
0 15000000
1 15000000
2 33000000
;
param:  t2p :=
1 -1
;

```

For simplicity of notation we omit the comments that will instead be available in the file. Instead the corresponding csv example is

```

delta_t;1;;;;;;;
rampup;70;;;;;;;
rampdown;70;;;;;;;
v_min;15000000;;;;;;;
v_max;33000000;;;;;;;
v_0;21080000;;;;;;;
v_T;21080000;;;;;;;
N_turbines;1;;;;;;;
N_pumps;1;;;;;;;
pump_activation_via_turbine;0;;;;;;;
theta_min;0;;;;;;;
s_max;0;;;;;;;
TURBINES qT_0 g_0 scT nOPT q_min q_max wT_init type plantT;1;0.00;0;75.00;3;8.400000;
42.000000;0.00;L;1;;
PUMPS qp_0 u_0 scP nOPP wP_init eP_init plantP;1;0.00;0;75.00;2;0.00;0.00;1;;;
R;2;;;;;;;
TURBINES NOPT Q_i;1;1;0.00;;;;;;;
;1;2;8.40;;;;;;;
;1;3;42.00;;;;;;;
TURBINES NOPT VOLUMES P_ir;1;1;1;0.000000;;;;;;;
;1;2;1;2.816118;;;;;;;
;1;3;1;23.272352;;;;;; ;1;1;2;0.000000;;;;;;;
;1;2;2;3.065391;;;;;;;
;1;3;2;25.906705;;;;;;;
PUMPS NOPP Q_u P_u;1;1;0.00;0.00;;;;;;;
;1;2;-26.98;-21.40;;;;;;;
k V;0;15000000.000000;;;;;;;
;1;15000000.000000;;;;;;;
;2;33000000.000000;;;;;;;
TURBINES t2p;1;1;;;;;;;

```

Instead in the case of a multi-reservoir format also some reservoir data are added, i.e., parameters  $J$ ,  $t2Up$ ,  $t2Dw$ , and  $tDelay$ . In addition, the reservoir index is made explicit in the quantities as described in Section 2. The same example in a mutiperiod format is

```

param J := 1;
param T := 3;
param: PERIODS: prices :=
1 35.45
2 33.06
3 32.01
;
param delta_t := 1;
param rampup := 70;
param rampdown := 70;
param: RESERVOIRS: inflows v_min v_max :=
1 1 2.48 15000000 33000000
1 2 2.31 15000000 33000000
1 3 2.17 15000000 33000000
;
param: v_0 v_T :=
1 21080000 21080000
;
param N_turbines := 1;
param N_pumps := 1;
param pump_activation_via_turbine := 0;
param theta_min := 0;
param s_max := 0;
param: TURBINES: qT_0 g_0 scT nOPT q_min q_max wT_init type plantT:=
1 0.00 0 75.00 16 8.40 42.00 0.00 L 1
;
param: PUMPS: qP_0 u_0 scP nOPP wP_init eP_init plantP:=
1 0.00 0 75.00 2 0.00 0.00 1
;
param R := 2;
param: Q_i :=
1 1 0.00
1 2 8.40
1 3 42.00
;
param: P_ir :=
1 1 1 0.000000
1 2 1 2.816118
1 16 1 23.272352
1 1 2 0.000000
1 2 2 3.065391
1 16 2 25.906705

```



```

;
param: Q_u P_u :=
1 1 0.00 0.00
1 2 -26.98 -21.40
;
param: V :=
1 0 15000000
1 1 15000000
1 2 33000000
;
param: t2p :=
1 -1
;
param: t2Up t2Dw tDelay:=
1 1 -1 0
;
while the corresponding csv format is
J;1;;;;;;;
T;3;;;;;;;
PERIODS prices;1;35.45;;;;;;;
;2;33.06;;;;;;;
;3;31.01;;;;;;;
delta_t;1;;;;;;;
rampup;70;;;;;;;
rampdown;70;;;;;;;
RESERVOIRS inflows v_min v_max;1;1;2.48;15000000;33000000;;;;;;;
;1;2;2.31;15000000;33000000;;;;;;;
;1;3;2.17;15000000;33000000;;;;;;;
RESERVOIR v_0 v_T;1;21080000;21080000;;;;;;;
N_turbines;1;;;;;;;
N_pumps;1;;;;;;;
pump_activation_via_turbine;0;;;;;;;
theta_min;0;;;;;;;
s_max;0;;;;;;;
TURBINES qT_0 g_0 scT nOPT q_min q_max wT_init type plantT;1;0.00;0;75.00;3;8.40;
42.00;0.00;L;1;;
PUMPS qP_0 u_0 scP nOPP wP_init eP_init plantP;1;0.00;0;75.00;2;0.00;0.00;1;;;;;
R;2;;;;;;;
TURBINES NOPT Q_i;1;1;0.00;;;;;;;
;1;2;8.40;;;;;;;
;1;3;42.00;;;;;;;
TURBINES NOPT VOLUMES P_ir;1;1;1;0.000000;;;;;;;
;1;2;1;2.816118;;;;;;;
;1;3;1;23.272352;;;;;;;
;1;1;2;0.000000;;;;;;;
;1;2;2;3.065391;;;;;;;

```

```

;1;3;2;25.906705;;;;;;
PUMPS NOPP Q_u P_u;1;1;0.00;0.00;;;;;;
;1;2;-26.98;-21.40;;;;;;
RERSERVOIR k V;0;15000000;;;;;;
;1;15000000;;;;;;
;2;33000000;;;;;;
TURBINES t2p;1;1;;;;;;
RESERVOIRS t2Up t2Dw tDelay;1;1;-1;0;;;;;;

```

The generator code is implemented in such a way that it allows the user to choose the extension of the output files whether only dat, only csv, or both. The generator offers the possibility of generating one or more single-reservoir or multi-reservoir instances. It is also possible to select the date of the data used from a predetermined subset, the number of periods to be considered, and the number of operational points. Once the number of o.p.s is chosen, the corresponding values are computed through a partition and then discretization of the corresponding production function. The original nonlinear production function used in the current version of the generator for every turbine is computed as in Borghetti et al. [5]. It is always possible to create personal instances by proposing one's own production function without altering the validity of the proposed format. Each instance is uniquely characterized and is therefore repeatable. At present, the generator allows instances to be generated only with 1-hour discretization since it is based on public data whose specific format is precisely 1 hour. It remains possible, however, to use a different time step duration  $\delta.t$  by providing other price and inflows data in the formats previously listed.

### 3.4 Stochastic extension

The data described so far relate to discrete formulations of problems involving energy management of cascading reservoir systems. In some countries managers are required to plan by considering the worst-case scenario, i.e., a deterministic formulation is often preferred. The generator does not currently generate stochastic instances, however, in some environments, it may be interesting to study such a formulation. For this purpose we also propose a format to describe the data with uncertainty. Specifically, the data in question are prices and inflows. Therefore in that case inflows and prices must have an additional index corresponding to the inflows and prices scenario, respectively. It is necessary to introduce the parameters

- $N_{inflows}$ : the number of realization scenarios of inflows.
- $N_{prices}$ : the number of realization scenarios of prices.

For each scenario of inflows it is necessary to introduce the parameter

- $prob_{inflows}$ : the probability between 0 and 1 of realization of a scenarios of inflows.

For each scenario of prices it is necessary to introduce the parameter

- *prob\_prices*: the probability between 0 and 1 of realization of a scenarios of prices.

An example of AMPL extension data of inflows and prices for a case with 3 periods, 2 scenarios of inflows with probability 0.4 and 0.6, 2 scenarios of prices with probability 0.3 and 0.7 is

```
# T is the number of time steps considered
param T := 3;
# N_inflows the number of realization scenarios of inflows
param N_inflows := 2;
# N_prices the number of realization scenarios of prices
param N_prices := 2;
# prob_inflows{ISCENARIOS} is the probability between 0 and 1 of realization
of a scenarios of inflows
param: ISCENARIOS: prob_inflows :=
1 0.4
2 0.6
;
# prob_prices{PSCENARIOS} the probability between 0 and 1 of realization
of a scenarios of prices
param: PSCENARIOS: prob_prices :=
1 0.3
2 0.7
;
# inflows{ISCENARIOS,RESERVOIRS,PERIODS} is the external (given) water
inflow for each time step [m3/s]
param: inflows :=
1 1 1 2.48
1 1 2 2.31
1 1 3 2.17
2 1 1 3.48
2 1 2 3.31
2 1 3 3.17
;
# prices{PSCENARIOS,PERIODS} is the unit price of the power generated/consumed
at each time step [euros/MWh]
param: prices :=
1 1 35.45
1 2 33.06
1 3 32.01
1 1 45.45
1 2 43.06
1 3 42.01
```

;

While the corresponding csv format is

```
T;3;;;;;;;;;;
N_inflows;2;;;;;;;;;;
N_prices;2;;;;;;;;;;
ISCENARIOS prob_inflows;1;0.4;;;;;;;;;;
;2;0.6;;;;;;;;;;
PSCENARIOS prob_prices;1;0.3;;;;;;;;;;
;2;0.7;;;;;;;;;;
ISCENARIOS PERIODS inflows;1;1;1;2.48;;;;;;;;;;
;1;1;2;2.31;;;;;;;;;;
;1;1;3;2.17;;;;;;;;;;
;2;1;1;3.48;;;;;;;;;;
;2;1;2;3.31;;;;;;;;;;
;2;1;3;3.17;;;;;;;;;;
PSCENARIOS PERIODS prices;1;1;35.45;;;;;;;;;;
;1;2;33.06;;;;;;;;;;
;1;3;32.01;;;;;;;;;;
;2;1;45.45;;;;;;;;;;
;2;2;43.06;;;;;;;;;;
;2;3;42.01;;;;;;;;;;
```

also shown in Table 4.

## 4 Hydro unit commitment models

In this section, we present some possible formulations of hydro unit commitment problems with a single reservoir. The production function in these problems is typically nonlinear so we present a MINLP formulation, however, we also propose some linearization variants through some MILP approximations. Starting from the input data the set of periods equivalent to *PERIODS* is defined as  $\mathcal{T}$ , a subset  $\mathcal{T}^* = \mathcal{T} \setminus \{1\}$ , and the set of utilities with the same cardinality of *TURBINES* and considering also the data for the corresponding *PUMPS* is defined as  $\mathcal{J}$ . In addition, some parameters necessary to the proposed models are calculated by pre-processing for every  $j \in \mathcal{J}$ , indeed combining the turbines

T	3			
N_inflows	2			
N_prices	2			
ISCENARIOS prob_inflows	1	0.4		
	2	0.6		
PSCENARIOS prob_prices	1	0.3		
	2	0.7		
ISCENARIOS PERIODS inflows	1	1	1	2.48
	1	1	2	2.31
	1	1	3	2.17
	2	1	1	3.48
	2	1	2	3.31
	2	1	3	3.17
PSCENARIOS PERIODS prices	1	1	35.45	
	1	2	33.06	
	1	3	32.01	
	2	1	45.45	
	2	2	43.06	
	2	3	42.01	

Table 4: Example of csv data of prices [ $\text{€}/MWh$ ] and inflows [ $m^3/s$ ] for 3 periods.

with the corresponding pumps, i.e.,

$$\underline{Q}_j = Q_{-ut2p_j, nOPP_{t2p_j}}, \quad (2)$$

$$\underline{P}_j = P_{-ut2p_j, nOPP_{t2p_j}}, \quad (3)$$

$$Q_{0j} = qT_{-0j} + qP_{-0_{t2p_j}}, \quad (4)$$

$$G_{0j} = g_{-0j}, \quad (5)$$

$$U_{0j} = u_{-0_{t2p_j}}, \quad (6)$$

$$W_j = wT_{-init_j}, \quad (7)$$

$$Z_j = wP_{-init_{t2p_j}}, \quad (8)$$

$$C_j = scT_j, \quad (9)$$

$$D_j = scP_{t2p_j}, \quad (10)$$

$$E_j = eP_{-init_{t2p_j}}, \quad (11)$$

where  $\underline{Q}_j$  and  $\underline{P}_j$  are the maximum water pumped and corresponding power consumed (as a negative value), respectively.  $Q_{0j}$  is the initial water flow considering both turbines and pumps,  $G_{0j}$  is the initial status of the turbine, and  $U_{0j}$  is the initial status of the pump corresponding to the turbine.  $W_j$  and  $Z_j$  are the water needed to start up the turbine or the corresponding pump, respectively.  $C_j$  and  $D_j$  are the start up costs of the turbine, the start up costs

and energy for the corresponding pump, respectively. Finally, the nonlinear production function  $f\{q_{jt}, v_{j0}\}$  based on the water flow  $q_{jt}$  of utility  $j$  at period  $t$  and the initial water volume is computed as

$$f\{q_{jt}, v_{j0}\} = \sum_{h \in H} \left( L_h q_{jt}^h \sum_{l \in H} (K_l v_{j0}^l - Llb - R0 q_{jt}^2) \right), \quad \forall t \in \mathcal{T}, \quad (12)$$

where  $H$  is the set of coefficients of the linear equation,  $R0$ ,  $Llb$ ,  $L_h$ , and  $K_h$  are some constants of the nonlinear function (see e.g., [5]). Note that the definition of  $f$  used above is simplified, as it assumes that the head-effect is negligible. We make this assumption for ease of notation of the approximation models introduced at the end of this section. Note that this is a simplification of the model that does not affect the generator.

#### 4.1 Common decision variables

As already mentioned one of the decision variables of the models is water flow  $q_{jt}$  of utility  $j$  at period  $t$ . In addition, the variable  $v_t$  represents the water volume at period  $t$ ,  $p_{jt}$  the power produced/consumed by utility  $j$  at period  $t$ , and  $s_t$ , i.e. the spillage at period  $t$ . Finally, a set of binary variables (1 on/true, 0 off/false) represents the status of the system, namely  $g_{jt}$ ,  $u_{jt}$ ,  $w_{jt}$ ,  $\tilde{w}_{jt}$ ,  $z_{jt}$ , and  $\tilde{z}_{jt}$  which represent, for the utility  $j$  at period  $t$ , the status for the turbine, the status of the pump, the case where the turbine is in shutting down phase, starting up phase, the case where the pump is in the shutting down phase, and the case where the pump is in starting up phase, respectively.

#### 4.2 MINLP formulation

As already mentioned the first presented model is a mixed integer nonlinear formulation variant of the model (MINLP) in Borghetti et al. [5]. For ease of notation we call in the following *delta.t* as  $\Delta t$ , *prices.t* as  $\Pi_t$ , and *inflows.t* as  $I_t$ . The volumes of water have been scaled dividing them by 1000, in order to

have a more stable and numerically robust solution. The model is therefore

$$\max \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} (\Delta t \Pi_t p_{jt} - C_j \tilde{w}_{jt} - (D_j + \Pi_t E_j) \tilde{z}_{jt}), \quad (13)$$

$$\text{s.t. } v_T \geq v_{-T}, \quad (14)$$

$$v_1 - v_0 - 3600 \Delta t \left( I_1 - \sum_{j \in \mathcal{J}} q_{jt} - s_t \right) = 0, \quad (15)$$

$$v_t - v_{t-1} - 3600 \Delta t \left( I_t - \sum_{j \in \mathcal{J}} q_{jt} - s_t \right) = 0, \quad \forall t \in \mathcal{T}^* \quad (16)$$

$$\sum_{j \in \mathcal{J}} q_{j1} - Q_{0j} + \text{rampdown} \geq 0, \quad (17)$$

$$\sum_{j \in \mathcal{J}} q_{jt} - q_{j(t-1)} + \text{rampdown} \geq 0, \quad \forall t \in \mathcal{T}^* \quad (18)$$

$$\sum_{j \in \mathcal{J}} q_{j1} - Q_{0j} - \text{rampup} \leq 0, \quad (19)$$

$$\sum_{j \in \mathcal{J}} q_{jt} - q_{j(t-1)} - \text{rampup} \leq 0, \quad \forall t \in \mathcal{T}^* \quad (20)$$

$$q_{jt} + (Q_j u_{jt} + q_{\text{min}_j} g_{jt}) \geq 0, \quad \forall j \in \mathcal{J}, t \in \mathcal{T} \quad (21)$$

$$q_{jt} + (Q_j u_{jt} + q_{\text{max}_j} g_{jt}) \leq 0, \quad \forall j \in \mathcal{J}, t \in \mathcal{T} \quad (22)$$

$$\sum_{j \in \mathcal{J}} q_{jt} + s_t - \text{theta\_min} \geq 0, \quad \forall t \in \mathcal{T} \quad (23)$$

$$s_t - \sum_{j \in \mathcal{J}} (W_j \tilde{w}_{jt} + Z_j \tilde{z}_{jt}) \geq 0, \quad \forall t \in \mathcal{T} \quad (24)$$

$$g_{jt} + u_{jt} \leq 1, \quad \forall j \in \mathcal{J}, t \in \mathcal{T} \quad (25)$$

$$g_{j1} - G_{0j} - (\tilde{w}_{j1} - w_{j1}) = 0, \quad (26)$$

$$g_{jt} - g_{j(t-1)} - (\tilde{w}_{jt} - w_{jt}) = 0, \quad \forall j \in \mathcal{J}, t \in \mathcal{T}^* \quad (27)$$

$$\tilde{w}_{jt} + w_{jt} \leq 1, \quad \forall j \in \mathcal{J}, t \in \mathcal{T} \quad (28)$$

$$u_{j1} - U_{0j} - (\tilde{z}_{j1} - z_{j1}) = 0, \quad (29)$$

$$u_{jt} - u_{j(t-1)} - (\tilde{z}_{jt} - z_{jt}) = 0, \quad \forall j \in \mathcal{J}, t \in \mathcal{T}^* \quad (30)$$

$$\tilde{z}_{jt} + z_{jt} \leq 1, \quad \forall j \in \mathcal{J}, t \in \mathcal{T} \quad (31)$$

$$p_{jt} - q_{jt} f\{q_{jt}, v-0\} g_{jt} \leq 0, \quad \forall j \in \mathcal{J}, t \in \mathcal{T}, \quad (32)$$

$$p_{jt} - \underline{P}_j u_{jt} + Pub_j g_{jt} \leq 0, \quad \forall j \in \mathcal{J}, t \in \mathcal{T} \quad (33)$$

$$\underline{Q}_j \leq q_{jt} \leq Qub_j, \quad \forall j \in \mathcal{J}, t \in \mathcal{T} \quad (34)$$

$$v\_min \leq v_t \leq v\_max, \quad \forall t \in \mathcal{T} \quad (35)$$

$$Plb_j \leq p_{jt} \leq Pub_j, \quad \forall j \in \mathcal{J}, t \in \mathcal{T} \quad (36)$$

$$0 \leq s_t \leq s\_max, \quad \forall t \in \mathcal{T} \quad (37)$$

$$g_{jt}, u_{jt}, w_{jt}, \tilde{w}_{jt}, z_{jt}, \tilde{z}_{jt} \in \{0, 1\} \quad \forall j \in \mathcal{J}, t \in \mathcal{T}, \quad (38)$$

where Constraints (14) set the final water volume at least to the desired target value at the end of the time horizon. Constraints (15)-(16) impose the water conservation within consecutive time periods. Constraints (17)-(20) limit the water flow variation. Constraints (21)-(22) establish the lower and upper bound of flow in each period. Constraints (23) establish a lower bound on the amount of water released in each period. Constraints (24)-(31) define the switch-on/switch-off rules of the turbines and pumps, imposing that if a turbine is on, then the corresponding pump should be off and vice versa. Constraints (32) represent the production function and Constraints (33) define an upper bound on the produced power. Finally, Constraints (34)-(37) bounds the continuous variables of the model.

### 4.3 MILP approximations

Three different MILP approximations of the production function  $p = \phi(q, v)$  have been introduced, namely convex combination model (CC), incremental model (Inc), and multiple choice model (MC), adapted to the application from [28]. In the presented linear approximations, the original nonlinear production function is replaced by a linear combination of the same. Each utility  $j$  is divided into  $S_j$  operating breakpoints, i.e.,  $k$  points at which there is a power  $\tilde{P}_{jk}$  corresponding to a water flow  $\tilde{Q}_{jk}$ . Assuming for each utility  $j$  a set  $S_j = \{0, \dots, S_j\}$  of breakpoints, and a subset  $S_j^* = S_j \setminus \{0\}$ , we introduce the new binary variables  $y_{jtk}$  (for  $j \in \mathcal{J}, t \in \mathcal{T}, k \in S_j$ ) which defines in which interval of production the values of  $q_{jt}$  and  $p_{jt}$  lie.



### 4.3.1 Convex combination model

A possible reformulation is a convex combination of the breakpoints. In addition to what already introduced also some continuous variables  $\lambda_{jtk}$ ,  $\mu_{jtk}$  are added to the model. Therefore, Constraints (21)-(22) and (32)-(33) are replaced by

$$p_{jt} \leq \sum_{k \in \mathcal{S}_j^*} \mu_{jtk} \tilde{P}_{j(k-1)} + \lambda_{jtk} \tilde{P}_{jk} + \underline{P}_j u_{jt}, \quad \forall j \in \mathcal{J}, t \in \mathcal{T} \quad (39)$$

$$\mu_{jtk} + \lambda_{jtk} = y_{jtk}, \quad \forall k \in \mathcal{S}_j^*, j \in \mathcal{J}, \\ t \in \mathcal{T} \quad (40)$$

$$\sum_{k \in \mathcal{S}_j^*} y_{jtk} = g_{jt}, \quad \forall j \in \mathcal{J}, t \in \mathcal{T} \quad (41)$$

$$q_{jt} + (\underline{Q}_j u_{jt} + \sum_{k \in \mathcal{S}_j^*} \mu_{jtk} \tilde{Q}_{j(k-1)} + \lambda_{jtk} \tilde{Q}_{jk}) \geq 0, \quad \forall j \in \mathcal{J}, t \in \mathcal{T} \quad (42)$$

$$q_{jt} + (\underline{Q}_j u_{jt} + \sum_{k \in \mathcal{S}_j^*} \mu_{jtk} \tilde{Q}_{j(k-1)} + \lambda_{jtk} \tilde{Q}_{jk}) \leq 0, \quad \forall j \in \mathcal{J}, t \in \mathcal{T} \quad (43)$$

$$0 \leq \mu_{jtk} \leq 1, \quad \forall k \in \mathcal{S}_j^*, j \in \mathcal{J}, \\ t \in \mathcal{T} \quad (44)$$

$$0 \leq \lambda_{jtk} \leq 1, \quad \forall k \in \mathcal{S}_j^*, j \in \mathcal{J}, \\ t \in \mathcal{T}. \quad (45)$$

### 4.3.2 Incremental model

Another possible reformulation is the incremental model introducing the continuous variable  $\delta_{jtk}$  and parameter  $\sigma_{jtk}$  computed as  $\frac{\tilde{P}_{jk} - \tilde{P}_{j(k-1)}}{\tilde{Q}_{jk} - \tilde{Q}_{j(k-1)}}$ . Therefore

Constraints (21)-(22) and (32)-(33) are replaced by

$$p_{jt} \leq \tilde{P}_{j0} y_{jt1} + \sum_{k \in \mathcal{S}_j^*} \sigma_{jtk} \delta_{jtk} + \underline{P}_j u_{jt}, \quad \forall j \in \mathcal{J}, t \in \mathcal{T} \quad (46)$$

$$\delta_{jtk} \leq (\tilde{Q}_{jk} - \tilde{Q}_{j(k-1)}) y_{jtk}, \quad \forall k \in \mathcal{S}_j^*, \\ j \in \mathcal{J}, t \in \mathcal{T} \quad (47)$$

$$\delta_{jtk} \geq (\tilde{Q}_{jk} - \tilde{Q}_{j(k-1)}) y_{j(k+1)t}, \quad \forall k \in \mathcal{S}_j^* \setminus \{S_j\}, j \in \mathcal{J}, \\ t \in \mathcal{T} \quad (48)$$

$$\delta_{jtS_j} \geq 0, \quad \forall j \in \mathcal{J}, t \in \mathcal{T} \quad (49)$$

$$y_{jt1} \leq g_{jt}, \quad \forall j \in \mathcal{J}, t \in \mathcal{T} \quad (50)$$

$$q_{jt} + (\underline{Q}_j u_{jt} + \tilde{Q}_{j0} g_{jt} \sum_{k \in \mathcal{S}_j^*} \delta_{jtk}) \geq 0, \quad \forall j \in \mathcal{J}, t \in \mathcal{T} \quad (51)$$

$$q_{jt} + (\underline{Q}_j u_{jt} + \tilde{Q}_{j0} g_{jt} \sum_{k \in \mathcal{S}_j^*} \delta_{jtk}) \leq 0, \quad \forall j \in \mathcal{J}, t \in \mathcal{T} \quad (52)$$

$$0 \leq \delta_{jtk} \leq \tilde{Q}_{jk} - \tilde{Q}_{j(k-1)}, \quad \forall k \in \mathcal{S}_j^*, j \in \mathcal{J}, \\ t \in \mathcal{T} \quad (53)$$

### 4.3.3 Multiple choice model

In the multiple choice reformulation, a new continuous variable  $\beta_{jtk}$  is introduced while a parameter  $\sigma_{jtk}$  is defined as Section 4.3.2. Therefore Constraints (21)-(22) and (32)-(33) are replaced by

$$p_{jt} \leq \sum_{k \in \mathcal{S}_j^*} \tilde{P}_{jk} y_{jtk} + \sigma_{jtk} (\beta_{jtk} - \tilde{Q}_{j(k-1)} y_{jtk}) + \underline{P}_j u_{jt}, \quad \forall j \in \mathcal{J}, t \in \mathcal{T} \quad (54)$$

$$\beta_{jtk} \leq \tilde{Q}_{jk} y_{jtk}, \quad \forall k \in \mathcal{S}_j^*, j \in \mathcal{J}, \\ t \in \mathcal{T} \quad (55)$$

$$\beta_{jtk} \geq \tilde{Q}_{j(k-1)} y_{jtk}, \quad \forall k \in \mathcal{S}_j^*, j \in \mathcal{J}, \\ t \in \mathcal{T} \quad (56)$$

$$\sum_{k \in \mathcal{S}_j^*} y_{jtk} = g_{jt}, \quad \forall j \in \mathcal{J}, t \in \mathcal{T} \quad (57)$$

$$q_{jt} + (\underline{Q}_j u_{jt} + \sum_{k \in \mathcal{S}_j^*} \beta_{jtk}) \geq 0, \quad \forall j \in \mathcal{J}, t \in \mathcal{T} \quad (58)$$

$$q_{jt} + (\underline{Q}_j u_{jt} + \sum_{k \in \mathcal{S}_j^*} \beta_{jtk}) \leq 0, \quad \forall j \in \mathcal{J}, t \in \mathcal{T}. \quad (59)$$

### 4.3.4 Considerations on the three MILP approximation models

Note that the three formulations introduced in this section are equivalent, i.e., they model the same approximation of the MINLP problem. Even though some

authors proved that some of them show some nice mathematical properties (see, for a complete review of alternative formulations, [28]), it is not possible to predict which one will be easier to solve with standard MILP solvers. Thus, it is interesting to introduce all of them and compare them from a practical viewpoint.

## 5 Computational Results

In this section, we present the results obtained by testing a relevant subset of single-reservoir instances that can be produced by the generator. The instances considered have been tested for an MINLP variant and also the three different MILP approximations of the production function CC, Inc, and MC. The MINLP and MILP models were coded in AMPL modeling language [1] and are available on the generator website, together with the instances used for computational results.

The set of instances has been generated by varying some parameters such as the reference period of the price and inflow data considering three different periods of the year as April, August, and December. The three different periods of the year are exemplary because seasonality leads to clear differences in climate and electricity consumption, therefore changing prices, demand, and availability of resources. In addition, three possible time horizons  $T$  have been considered, i.e., 24, 48, and 72 hours, respectively, and finally, also different combinations of operational points (*o.p.*) have been considered. We have picked: 3, 10, 16, 23, and 30.

All experiments are performed on a single machine equipped with an Intel Xeon E5649 processor clocked at 2.53 GHz and 50 GB RAM. We solved the MINLPs with the exact solvers Baron [22] and Couenne [3] or with the heuristic solvers Bonmin [4] and Knitro [6]. Note that they are heuristic because our MINLP is nonconvex. The MILPs models were solved with the IBM ILOG CPLEX 12.6 solver [12]. A time limit (t.l.) of one hour was imposed to all the MINLP and MILP solvers.

We start analyzing the solution of the MINLP problem, which is a more realistic formulation compared to the MILP approximation. In Tables 5 and 6 we report the results of the four MINLP solvers on the 9 instances, in particular, the best solution value and the CPU time, respectively. The first two columns of each of the tables represent the characteristics of the considered instance, namely the considered month (“m”) and time horizon (“T”). The best entry is highlighted in bold.

Table 5 clearly shows the good performance of Baron in finding the best solution for the majority of the instances except for i. August instance with a time horizon of 24h, where Baron wrongly proves a global optimum which is worse than the best solution found by Knitro; ii. the two largest instances for December, where Bonmin finds a better solution. However, from Table 6 we see that Baron and Couenne, the two exact solvers, almost always hit the time limit. When the time limit is not reached, it is because of some error: as explained before,

m	T	KNITRO	Baron	Bonmin	Couenne
4	24	4,812.01	<b>5,087.73</b>	4,496.69	3,784.17
4	48	9,476.48	<b>10,053.20</b>	9,387.81	5,010.39
4	72	14,257.35	<b>14,703.75</b>	14,120.03	7,009.04
8	24	<b>216.82</b>	<i>120.63</i>	176.12	<i>N/A</i>
8	48	919.35	<b>1,285.98</b>	639.91	<i>N/A</i>
8	72	1,850.38	<b>1,971.93</b>	1,130.83	<i>N/A</i>
12	24	<b>4,776.82</b>	<b>4,776.82</b>	4,075.47	<i>N/A</i>
12	48	11,788.25	-	<b>15,271.65</b>	4,122.74
12	72	13,392.96	19,023.91	<b>19,176.64</b>	6,718.91

Table 5: Comparison among MINLP solvers for the different instances: optimal profit value (€k)

m	T	KNITRO	Baron	Bonmin	Couenne
4	24	<b>1.99</b>	t.l.	13.01	t.l.
4	48	316.58	t.l.	<b>73.14</b>	t.l.
4	72	348.44	t.l.	<b>227.04</b>	t.l.
8	24	<b>2.90</b>	3.23	5.99	2.79
8	48	<b>17.83</b>	t.l.	18.76	7.70
8	72	111.85	t.l.	<b>40.06</b>	16.18
12	24	3.30	t.l.	<b>1.01</b>	17.68
12	48	121.52	t.l.	<b>16.47</b>	t.l.
12	72	811.28	t.l.	<b>33.29</b>	t.l.

Table 6: Comparison among MINLP solvers for the different instances: CPU time (s)

Baron proves a wrong global optimum in the case of  $m = 8$  and  $T = 24$ . In the four cases where Couenne terminates in less than one hour, it returns an “infeasible problem” message, which is wrong as well because the other solvers could find a feasible solution. This shows that the problem is challenging when considering the nonlinear production function. Thus, the heuristic solvers show better CPU times at the cost of losing some solution quality.

We now move to the analysis of the solution of the MILP approximations. In this case as well, the solution found is only heuristic. However, it is well known that MILP solvers are in practice much more efficient than the MINLP ones, thus it is worth considering such approximations.

Table 7 shows the computing time [s] for three different MILP formulations, testing the same set of instances for the MINLP case, but considering as well different numbers of o.p. For every data reference period, the best performing formulation among the three is in bold. The Inc formulation seems to be the most performing. Note that the difficulty of the instances is not necessarily proportional to its size or the number of binary variables (which is proportional to  $T$  and o.p.). For example, the case of April,  $T = 72$  hours, and 30 o.p., the largest considered instance, seems to be easier than smaller instances like April,  $T = 72$ , and 10 o.p. Note as well that, as expected, the CPU time needed to solve the MILP formulations is much shorter than the one needed for the MINLP ones. Moreover, CPLEX never hits the time limit when solving Inc.

T	o.p.	April			August			December		
		CC	Inc	MC	CC	Inc	MC	CC	Inc	MC
24	3	0.08	<b>0.04</b>	0.05	0.04	0.04	<b>0.04</b>	0.04	0.03	<b>0.03</b>
24	10	3.89	<b>0.25</b>	0.68	0.07	<b>0.05</b>	0.06	0.12	<b>0.07</b>	0.12
24	16	6.81	<b>0.35</b>	1.27	0.14	<b>0.07</b>	<b>0.07</b>	0.19	<b>0.14</b>	0.21
24	23	11.48	<b>0.49</b>	1.86	0.17	<b>0.07</b>	0.09	0.24	<b>0.15</b>	0.30
24	30	17.17	<b>0.64</b>	3.02	0.20	<b>0.10</b>	0.12	0.40	<b>0.23</b>	0.36
48	3	5.98	5.75	<b>4.64</b>	0.05	<b>0.04</b>	0.04	0.07	<b>0.05</b>	0.06
48	10	0.09	<b>0.05</b>	0.12	0.19	<b>0.09</b>	0.14	0.07	<b>0.03</b>	0.09
48	16	21.69	<b>0.95</b>	8.28	0.33	<b>0.19</b>	0.25	<b>0.09</b>	0.10	0.11
48	23	4.93	<b>0.49</b>	0.86	0.38	<b>0.22</b>	0.36	<b>0.12</b>	<b>0.12</b>	0.19
48	30	0.52	<b>0.25</b>	1.31	0.79	<b>0.33</b>	0.60	0.16	<b>0.15</b>	0.19
72	3	416.38	227.99	<b>215.04</b>	0.06	<b>0.06</b>	0.07	0.12	<b>0.09</b>	0.11
72	10	t.l.	<b>1532.47</b>	t.l.	0.26	<b>0.15</b>	0.36	2.05	<b>0.20</b>	0.93
72	16	2.99	<b>0.66</b>	1.47	0.84	<b>0.45</b>	0.78	12.01	<b>0.41</b>	1.34
72	23	t.l.	<b>59.28</b>	t.l.	1.46	<b>0.67</b>	1.41	51.47	<b>0.41</b>	16.70
72	30	8.1	<b>0.90</b>	3.46	2.27	<b>0.99</b>	2.03	116.34	<b>0.58</b>	51.42

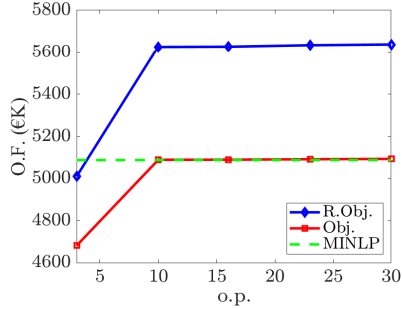
Table 7: CPU time (s) using the approaches CC, Inc, MC for the different instances.

Finally, we remind the reader that the three formulations are equivalent, thus they find the same optimal solution, for a given number of o.p.

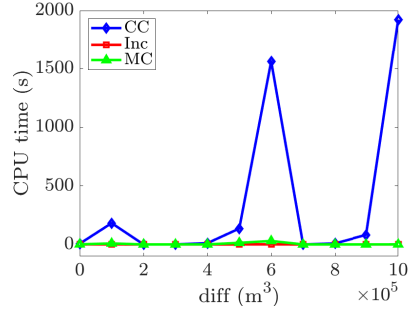
Additional comparison has been performed with the original MINLP formulation solved with Baron. In Figure 3a we present some examples of optimal values obtained with MINLP and MILP approaches within the time limit, the time horizon of 24 hours, and different numbers of operational points, which determine the accuracy of the MILP approximation. The green dotted line “MINLP” represents indeed the best solution found with the MINLP model (even if with a gap > 300%), the blue line “R.Opt.” are the optimal solutions obtained with any of the MILP formulations which seem to overestimate, while the red line “Opt.” represents the values computed using MILP solutions within the original MINLP objective function, i.e., the projection of the linear solution into the nonlinear formulation. Using the optimized solution of the linear relaxations, it is always possible to compute the value of the objective function with nonlinear power formulation based on the obtained solution. As expected the accuracy of the approximated production function  $p = \phi(q, v)$  is increasing as the number of o.p. and using only 10 o.p. it is possible to obtain a practically perfect approximation by calculating a posteriori the nonlinear objective function of the optimal solution.

Instead, Figure 3b shows the computing time of some instances as a function of the difference between the initial water volume and the target water volume, which ranges from 0 to 1 [ $hm^3$ ]. In particular, the considered instances have been generated fixing the parameters to 16 o.p., data were taken from April, and a 24 hours time horizon was picked. Again, MINLP results are not mentioned as the time limit is once more reached. The result shows that the aforementioned difference affects the complexity of the problem as they clearly influence the possible values of volume resources.

Finally, we provide a detailed comparison of the objective function value pro-



(a) Comparison of objective functions of the relaxed models and computed using MILP solutions within the original MINLP objective function, of instances in function of the different o.p. 3, 10, 16, 23, and 30, with data from April, and 24 hours horizon.



(b) Comparison of CPU solving time of instances in function of the difference between the initial water volume and the target water volume  $\text{diff} (m^3)$  with 16 o.p. and 24 hours horizon.

Figure 3: Comparison of objective function value and CPU time.

vided by the MILP and the MINLP formulations in Table 8.

m	T	3	10	16	23	30	best sol MINLP	best UB
4	24	4,681.45	5,089.11	5,089.40	5,091.50	<b>5,093.08</b>	5,087.74	20,202.67
4	48	9,105.78	<b>10,289.96</b>	10,264.88	10,281.78	<b>10,289.96</b>	10,053.20	43,691.41
4	72	13,319.27	15,187.22	<b>15,240.99</b>	15,221.61	<b>15,240.99</b>	14,703.75	58,568.21
8	24	<b>216.82</b>	<b>216.82</b>	<b>216.82</b>	<b>216.82</b>	<b>216.82</b>	<b>216.82</b>	N/A
8	48	<b>1,285.98</b>	<b>1,285.98</b>	<b>1,285.98</b>	<b>1,285.98</b>	<b>1,285.98</b>	<b>1,285.98</b>	2,032.73
8	72	1,971.93	1,971.93	<b>2,028.87</b>	<b>2,028.87</b>	<b>2,028.87</b>	1,971.93	7,131.61
12	24	4,426.82	4,775.23	4,776.33	4,776.78	4,776.31	<b>4,776.82</b>	10,731.09
12	48	13,863.24	15,212.33	15,300.27	15,322.00	<b>15,334.12</b>	15,271.65	33,563.56
12	72	16,971.23	19,273.00	19,314.59	19,322.43	<b>19,322.78</b>	19,176.64	44,879.98

Table 8: Comparison among MINLP solvers for the different instances (€k)

Columns  $m$  and  $T$  refer to the month and time horizon characterizing the instances. The next 5 columns show the profit value of the best solution found by the MILP formulations for 3, 10, 16, 23, 30 o.p., respectively. The great majority of the time, a larger number of o.p. implies a better objective function value, but it is not systematic (see, for example, the value for  $m = 4$  and  $T = 48$ ). This is normal because the MILP formulations are just approximations of the MINLP formulation. In the last two columns, we reported: “best sol MINLP”, i.e., the best solution value found by any of the MINLP solvers; “best UB” the best upper bound found by Baron and Couenne. We can observe that the best solution found by the MINLP solvers is never better than the one found by the MILP solvers for a number of o.p.  $\geq 16$  (besides  $m = 12$  and  $T = 24$  if we consider the decimals). Furthermore, the best upper bound, when available, is always very far from the best known solution. We suspect that the relaxation

solved by Baron/Couenne to find the upper bound is very weak. To conclude, the MILP formulations seem to be a good option when one wishes to have a good trade off between solution quality and short CPU time.

## 6 Perspectives and Conclusions

The presented instance generator was shown to yield already non-trivial instances when considering only a single reservoir. Multiple-reservoir instances also contain information about the cascading structure. In the future, we are planning to extend the generator to other existing structures and drainage basins. In addition, the tool can be used with different price and inflow data with minor modifications. For instance, a possible future extension could be to allow the generator to provide automatically also data with half-hourly and quarter-hourly periods and also to handle exceptions such as incomplete data. This research benefited from the support of the FMJH Program PGM0 and from the support of EDF.

## References

- [1] AMPL Optimization Inc. <https://ampl.com/>.
- [2] H. E. Beck, A. I. J. M. van Dijk, V. Levizzani, J. Schellekens, D. G. Miralles, B. Martens, and A. de Roo. Mswep: 3-hourly 0.25 global gridded precipitation (1979-2015) by merging gauge, satellite, and reanalysis data. *Hydrology and Earth System Sciences*, 21(1):589–615, 2017.
- [3] P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wächter. Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods and Software*, 24(4):597–634, 2009.
- [4] P. Bonami, L. Biegler, A.R. Conn, G. Cornuéjols, I.E. Grossmann, C.D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. Wächter. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization*, 5:186–204, 2008.
- [5] A. Borghetti, C. D’Ambrosio, A. Lodi, and S. Martello. A MILP approach for short-term hydro scheduling and unit commitment with head-dependent reservoir. *IEEE Transactions on Power Systems*, 23(3):1115–1124, 2008.
- [6] R. Byrd, J. Nocedal, and R. Waltz. Knitro: An integrated package for nonlinear optimization. In G. Di Pillo and M. Roma, editors, *Large-Scale Nonlinear Optimization*, pages 35–60. Springer, 2006.
- [7] E. Dutra, G. Balsamo, J-C. Calvet, S. Munier, S. Burke, G. Fink, A. Martínez de la Torre, R. Beek, A. de Roo, and J. Polcher. Report on the improved water resources reanalysis. Technical report, 06 2017.

- [8] ENEL. Enel. <https://www.enel.it/>.
- [9] E.C. Finardi and E.L. Da Silva. Solving the hydro unit commitment problem via dual decomposition and sequential quadratic programming. *IEEE Transactions on Power Systems*, 21(2):835–844, 2006.
- [10] E.C. Finardi, E.L. Da Silva, and C.A. Sagastizábal. Solving the unit commitment problem of hydropower plants via Lagrangian relaxation and sequential quadratic programming. *Computational & Applied Mathematics*, 24(3):317–341, 2005.
- [11] E.C. Finardi, F. Y. K. Takigawa, and B. H. Brito. Assessing solution quality and computational performance in the hydro unit commitment problem considering different mathematical programming approaches. *Electric Power Systems Research*, 136:212–222, 2016.
- [12] IBM. <https://www.ibm.com/fr-fr/products/ilog-cplex-optimization-studio>.
- [13] J. Kong, H. I. Skjelbred, and O. B. Fosso. An overview on formulations and optimization methods for the unit-based short-term hydro scheduling problem. *Electrical Power Systems Research*, 178(1):1–14, 2020.
- [14] Jiehong Kong, Hans Ivar Skjelbred, and Olav Bjarte Fosso. An overview on formulations and optimization methods for the unit-based short-term hydro scheduling problem. *Electric Power Systems Research*, 178:106027, 2020.
- [15] G. Krinner, N. Viovy, N. de Noblet-Ducoudré, J. Ogée, J. Polcher, P. Friedlingstein, Ph. Ciais, S. Sitch, and I. C. Prentice. A dynamic global vegetation model for studies of the coupled atmosphere-biosphere system. *Global Biogeochemical Cycles*, 19(1), 2005.
- [16] A. Lodi, E. Malaguti, G. Nannicini, and D. Thomopulos. Nonlinear chance-constrained problems with applications to hydro scheduling. *Mathematical Programming*, 11 2019.
- [17] T. Nguyen-Quang, J. Polcher, A. Ducharne, T. Arsouze, X. Zhou, A. Schneider, and L. Fita. Orchidee-routing: revising the river routing scheme using a high-resolution hydrological database. 2018.
- [18] Hans Ivar Skjelbred. Shop (short-term hydro optimization program). <https://www.sintef.no/en/software/shop/>.
- [19] M. Tahanan, W. van Ackooij, A. Frangioni, and F. Lacalandra. Large-scale unit commitment under uncertainty: a literature survey. *4OR*, 13(2):115–171, 2015.
- [20] R. Taktak and C. d’Ambrosio. An overview on mathematical programming approaches for the deterministic unit commitment problem in hydro valleys. *Energy Systems*, 8(1):57–79, 2017.



- [21] Elvis Del Tedesco. Progettodighe. <https://www.progettodighe.it>.
- [22] The Optimization Firm. <https://minlp.com/baron>.
- [23] W. van Ackooij, C. D'Ambrosio, L. Liberti, R. Taktak, D. Thomopulos, and S. Toubaline. Shortest path problem variants for the hydro unit commitment problem. *Electronic Notes in Discrete Mathematics*, 69:309–316, 2018.
- [24] W. van Ackooij, I. Danti Lopez, A. Frangioni, F. Lacalandra, and M. Tahanan. Large-scale unit commitment under uncertainty: an updated literature survey. *Annals of Operations Research*, 271(1):11–85, 2018.
- [25] W. van Ackooij and W. de Oliveira. Nonsmooth DC-constrained optimization: constraint qualification and minimizing methodologies. *Optimization Methods and Software*, 34(4):890–920, 2019.
- [26] W. van Ackooij, E. C. Finardi, and G. Matiussi Ramalho. An exact solution method for the hydrothermal unit commitment under wind power uncertainty with joint probability constraints. *IEEE Transactions on Power Systems*, 33(6):6487–6500, 2018.
- [27] W. van Ackooij, R. Henrion, A. Möller, and R. Zorgati. Joint chance constrained programming for hydro reservoir management. *Optimization and Engineering*, 15:509–531, 2014.
- [28] J.P. Vielma, S. Ahmed, and G. L. Nemhauser. Mixed-integer models for nonseparable piecewise-linear optimization: Unifying framework and extensions. *Operations Research*, 58:303–315, 2010.