# Configuration of Hydro Power Plant Mathematical Models

## Research in Progress Paper

Michael Barry[1], Moritz Schillinger[2], Hannes Weigt[2], and René Schumann[1]

[1] Smart Infrastructure Laboratory, HES-SO Valais / Wallis ,
Rue de Technopôle 3, 3960 Sierre, Switzerland
michael.barry@hevs.ch | rene.schumann@hevs.ch
[2] Forschungsstelle Nachhaltige Energie- und Wasserversorgung, University Basel,
Peter Merian-Weg 6, 4002 Basel, Switzerland
moritz.schillinger@unibas.ch | hannes.weigt@unibas.ch

**Abstract.** The ongoing energy transition towards large shares of renewable generation poses challenges for hydro power producers. We revisit the problem of optimising the operation of hydro power plants using mathematical modelling, but utilising computer science concepts in the design of the models and configuration of these models. In particular we use a configuration process based on the SPEA2 evolutionary algorithm to identify module based configurations of the model and explore the trade-off between the scale of the model and it's runtime. It is hoped that such methods can assist in identifying configurations that are the best fit in terms of runtime, realism and accuracy.

## 1 Introduction

Due to the energy transition throughout Europe there is an ongoing shift from using conventional energy sources such as fossil or nuclear to renewables such as photo-voltaic or wind power. However, such power sources are intermittent due to weather conditions and therefore create larger fluctuations in the energy system. These fluctuations can also be seen in the energy prices with a significant decline of the average price levels and a flattening of daily price peaks due to solar injections. Hydro power (HP) is seen as a viable option for stabilizing the system, as it can in effect act as a battery. Even without pump storage, HP has the ability to release its water reservoirs when other renewable energy sources cannot produce. This allows a hydro plant to stabilise the system as well as profit of high energy prices during times of low renewable feed-in.

HP is a mature technology with over a century of utilization history and is therefore well understood, including it's optimisation. Highly accurate mathematical models have already existed for some time. However, recently there have been several factors contributing to complicating the problem and therefore forcing us to revise our methods.

The first factor is that due to the less predicable power production, and the resulting fluctuations in the market prices, the operation plans needs to be quickly adapted. As planning relevant information becomes only available short term, the resulting method for operation planning has to have low runtimes. Also markets, like the intra-day market emerged, that can be a highly lucrative market for HP, but only allows for a short time frame for the required calculations.

In addition to having less time, the problem has also become larger, as the intra-day market works on 15 minute time slots, with a possible reductions to 10 minutes. Using such smaller time slots greatly increases the complexity of the problem and the amount of data needed. In addition, further environmental constraints have to be integrated also increasing the problems size.

Even though optimisation becomes harder, it also becomes more important. Like other conventional power plants also HP plants struggle with the currently low energy prices. In particular as they have to pay fees to use the water and are exempted from most green energy benefits.

One solution might seem to shift from mathematical modelling to more heuristic orientated methods. But in this field mathematical modelling is well established, and existing models should stay in use, when possible. In addition, most market models use mathematical modelling and must be compatible.

However, there are many aspects in mathematical modelling that can benefit from fundamental concepts in computer science. This includes design, development and deployment aspects. In this paper we focus on the configuration of the mathematical models. Current models are problem specific and are hard to solve. However, in some instances it can be beneficial to use a lighter model with a lower runtime, so that more up-to-date data can be used. We will provide here an approach towards a more flexible model that can adjust its scale.

Our idea bases on the assumption that a mathematical model is based on different modules describing aspects of the problem. For each aspect alternative modules can exist. By combining different modules a complete model can be created. The combination of modules is the *configuration* of the model. Based on the idea that mathematical models are *wrong, but some are useful.* [3], we are going to investigate the trade-off between usefulness of a model and its resulting computational complexity. Therefore, we will investigating the search space of all different configurations of these models in terms of scale and the required time to solve them. To investigate this question we propose an easy to implement method to show the trade-off between the scale of the model and it's runtime by investigating the Pareto front. The Pareto front is expected to be interesting both in a research and practical sense.

The optimal configurations and the pattern in which they can be observed may be of research interest, as they may point to an easy-hard-easy pattern [6]. The aim is to identify patterns in the optimal configurations within the Pareto front for different case studies (different hydro plants) to define general guidelines that can assist in designing models in the future.

This is a research in progress paper. First we discuss some related literature. In Section 3 the problem definition is presented, followed by our approach in Section 4. We then provide first results (Section 5), and finally outline our future work and expectation.

## 2 Background

Using mathematical models for the planning of the ideal operation of a hydro plant is a well known and tested method. This is the case for both single site models [5, 7] and cascading (multisite) models [4, 9, 1]. Even large models, such as the HP plant system in Brazil consisting of 150 hydro plants, can be modelled using this method [8]. However, there are, to our knowledge, no other works that aims to use a modular design to develop these models and utilise this design as part of a configuration process to ascertain ideal configurations. As for the configuration of the model, it is a question of how simple the model should be. Operations research is very familiar to the concept that *Essentially, all models are wrong, but some are useful.* [3] as well as with the somewhat opposing principle of Occam's razor [2], stating that with *competing hypotheses that predict equally well, the one with the fewest assumptions should be selected.* For the field of operations research, it is often difficult to justify a model to be simple but accurate enough expect through practical testing. In this paper, we attempt to create a more systematic approach. In addition, through identifying the Pareto front, we hope to find an easy-hard-easy pattern [6].

The configuration is multi-objective as we aim to minimise the required run-time to solve the model and reduce the scale of the model. Multi-objective problems usually consider the Pareto front as a solution, which are proved to be NP-hard to compute. [10] Therefore heuristics are commonly used, of which evolutionary methods have proven to be effective off-the shelf algorithms [10].

## 3 Problem definition

### 3.1 Optimisation Problem

The optimisation problem of a HP plant can be defined in relatively simple terms and is similar to a mathematical representation of a battery. However, we must consider that we have the possibility of trading on several markets. In this Paper we consider the optimisation problem in basic form for simplicity. However, we consider the problem to be scalable, as many technical, environmental and market constraints can be added. The basic form is shown below:

$$\max. \sum_{i,m} c_{i,m} P_{i,m} \tag{1}$$

$$P_{i,m} = R_{i,m} \alpha \tag{2}$$

$$S_i = S_{i-1} + I_i - \sum_m R_{i,m} \tag{3}$$

$$S_i \leq S_{\max} \tag{4}$$

$$S_i \geq S_{\min} \tag{5}$$

Where $c_{i,m}$ is the price at time interval $i$ for market $m$, $P_{i,m}$ is the produced energy for time interval $i$ and market $m$, $R_{i,m}$ is the water released from the reservoir at time interval $i$ for market $m$, $\alpha$ is the efficiency of the turbine (the amount of energy produced per water used), $S_i$ is the storage level at time interval $i$, $I_i$ is the inflow of water into the reservoir at time interval $i$, $S_{\max}$ is the maximal and $S_{\min}$ is the minimum storage level of the reservoir. This model is of course a strict simplication, it lacks technical and environmental constraints.

### 3.2 Configuration Problem

Below we describe the configuration of the model used to solve the optimisation problem defined above. Each configuration is defined by which markets are used. Using all markets increases the scale of the model and therefore directly affects the time required to solve it. The configuration problem can be summarised as finding the Pareto front of this trade off. The Problem is defined formally below:

Given a configuration $x$ in the set of all possible configurations $A$ and functions $f_i(x)$ for each objective i, x is said to be Pareto optimal if no other member $x*$ of $A$ dominates $x$. $x*$ dominates x if the following is true:

$$f_i(x) \leq f_i(x^*) \text{for all } i \text{ where} f_i(x) < f_i(x^*) \text{for at least one objective } i \tag{6}$$

For the configuration of the HP model, there are two objective $f_r(x)$ and $f_s(x)$ where $f_r(x)$ represents the runtime of configuration x and $f_s(x)$ represents the scale of the model in configuration x.

## 4 Method

This section describes our own approach to address the previously described problems, utilising the General mathematical Modelling System (GAMS), the IBM CPLEX solver and the SPEA2 evolutionary algorithm [11].

### 4.1 Mathematical Model Design

As previously described, we aim at creating scalable models that can be easily configured. An object orientated inspired design was used to map functions to features, resulting in a group of functions, or module, to represent separate functionalities contained within a separate file. A main file containing a list of import statements can be adjusted to simply exclude a file and therefore it's feature. For example, each function for the intra-day market is contained in a module and therefore it is possible to switch off trading on the intra-day market by excluding the intra-day market module. As the model grows, new modules are added that either replace modules or compliments them. Additional constraint can be written in new modules and added, for instance a new turbine for a new case study can be written and then added in a simulation for the new case study. The entire model is implemented using GAMS and once configured, CPLEX is used to solve it. This modular design has many benefits similar to object orientated design, including abstraction, mapping modules or real life objects, better maintainability and deployment.

**Data**: $N$: population size, $\overline{N}$: elite population size, $T$ Max number of generations
**Result**: $\overline{P}_t$ the non-dominated set

Initialise population $P_0$ with randomly generated individuals and empty set $\overline{P}_0$
Set t=0
**while** $t > T$ **do**
  **for all** x in $P_0$ and $\overline{P}_0$ **do**
    $f(x) = N - N_d$ where $f(x)$ is the fitness function and $N_d$ is the number by which the individual in dominated by
  **end for**
  Move non-dominated individuals from $P_t$ and $\overline{P}_t$ to $\overline{P}_{t+1}$.
  **if** size of $\overline{P}_{t+1} > \overline{N}$ **then**
    use clustering method to reduce the size of $\overline{P}_{t+1}$
  **end if**
  **for all** x in $P_t$ and $\overline{P}_t$ **do**
    move $P_t$ to $P_{t+1}$ based on roulette wheel selection
  **end for**
  **while** Size of $P_{t+1} < N$ **do**
    breed $x^*$ from $\overline{P}_t$ and $P_t$
    add $x^*$ to $P_{t+1}$
  **end while**
  t = t + 1
**end while**
**return** $\overline{P}_T$

**Algorithm 1:** SPEA2 algorithm

### 4.2 Configuration

Each set of modules used is considered to be a configuration. A list of modules that are required for the basic model to run for a particular case study is used to ensure those models are always switched on and the remaining modules are considered by the configuration process. The configuration process is based on the SPEA2 algorithm. It uses Pareto domination in it's fitness function, a separate population of non-dominated individuals to implement elitism and a clustering algorithm to stop convergence. Details are given in the listing of Algorithm 1.

As mentioned, we use the runtime and scale of the model as objectives. As the runtime is dependent on the hardware used and other software running, we use the CPLEX ticks as a platform independent measure of runtime. We use the number of modules as a preliminary measure of the scale of the model. Although this is a relative simple method it has shown to be fairly dependable. In the future, we plan to exchange this measure with more sophisticated ones.

We have chosen an evolutionary algorithm for the following reason. First, a population based approach is well suited for finding the Pareto front, as an entire set of Pareto optimal solutions are contained in the elite population, reducing the number of requiring reruns. Second, an iterative population based approach effectively investigating an unknown search space, as it provides the stability of the scale measurement by identifying if related individuals also have similar runtimes. In addition, evolutionary algorithms are highly configurable, allowing us to update separate components, such as the initialisation of the population, to achieve better performance.

## 5 Initial results

### 5.1 Optimisation

Some results of solving the mathematical model are shown in Fig. 1. We added markets in order, therefore a model with 4 markets has markets 1-4 activated. Each market has a separate implementation including the day-ahead market, intra-day market, primary and secondary reserve market, positive and negative tertiary reserve market. These markets modules are test implementations and contain fictive test data. The modules are complementary and can be combined in one model.

In general, it shows how increasing the number of markets that the model is able to trade on also increases the revenue, showing that there is a clear profit gain in being able to trade on all markets. However, market 2 appears to never be favourable which is why trading in 1 or two markets in Fig. 1 shows no difference. Trading in all 6 markets or all expect market 2 also have the same results (not shown in Fig. 1). Thus, it may be of use to exclude market 2 from the model.

### 5.2 Configuration

In this section we present our initial results. These results are only viable as a proof of concept, but are not yet complete, mostly due to the fact that the model is still at an early stage and therefore is simply not large enough yet to create a large enough search space for the evolutionary algorithm. For this section we use a small scale model as a proof of concept, with simple implementations of

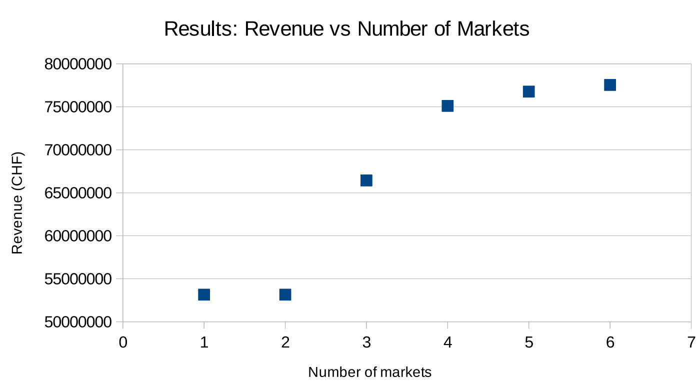


**Fig. 1.** Initial results of the configuration process, showing the pareto front

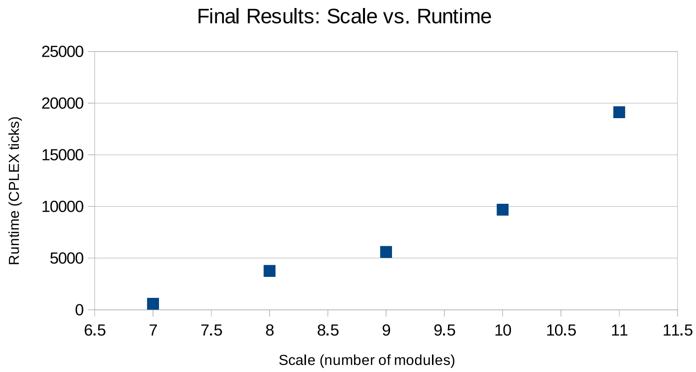


**Fig. 2.** Initial results of the configuration process, showing the Pareto front

the technical aspects and several markets to choose from. The model is scalable only through the choice of which and how many markets to trade on. The evolutionary algorithm may seem obsolete in these initial results due to the small scale, especially as the clustering algorithm is never activated due to the limited number of possible Pareto optimal solutions. Fig. 2 shows the final results, outlining the Pareto front. Fig. 2 demonstrates a clear relationship between the scale measurement and runtime, which demonstrates the feasibility of using the number of modules as a measurement of scale. It shows that configurations with similar scale also have a similar runtime and that there is a proportional relationship between the runtime and the scale. However, more details about how stable and what type of relationship exists will only be obtained once a large model is implemented.

## 6 Future work and expectations

The research project is still in an early stage. In the next steps we will focus on expanding the mathematical model, greatly increasing the search space of the configuration problem. As we are going to apply our approach in real-world case studies, competing with industry standards, we have to implement more technical and environmental constraints and more sophisticated market models. In total the project will contain two case studies, first a single HP plant, and later cascading HP plants, for which operation planning has to be done in one planning process, due to their physical dependencies.

Beside increasing the complexity of the model, we also need to revise our measurements for the size of the problem. The currently used number of modules is just an intermediate step, and more sophisticated models needs to be defined.

Once the model is of a larger scale, we will improve the configuration. There are large improvements that can be done for the evolutionary algorithm, especially once we understand the search space better. Knowledge of previous runs can be used in the initialisation of the population, reducing the time to find the Pareto front. To assist in the population spreading along the Pareto front a modified mutation operator can be used to favour mutation to a larger or smaller scale model. Additionally, the fitness evaluating requires to solve the model and therefore is time consuming. To speed-up evaluation, a hash map can be used to look-up solutions instead of recomputing them.

We also need to analyse in more depth the search space, especially whether the stability we observed above also exists in larger models, and whether the simple relationship between scale and runtime remains or if fluctuations or even easy-hard-easy curves can be observed. This insights can be used to identify areas that promise to have a low runtime despite a relatively large model.

## Acknowledgments

## References

1. Alfieri, L., Perona, P., Burlando, P.: Optimal water allocation for an alpine hydropower system under changing scenarios. Water Resources Management 20(5), 761–778 (2006)
2. Blumer, A., Ehrenfeucht, A., Haussler, D., Warmuth, M.K.: Occam's razor. Information Processing Letters 24(6), 377 – 380 (1987)
3. Box, G.E., Draper, N.R.: Empirical model-building and response surfaces. John Wiley & Sons (1987)
4. Guo, S., Chen, J., Li, Y., Liu, P., Li, T.: Joint operation of the multi-reservoir system of the three gorges and the qingjiang cascade reservoirs. Energies 4(7), 1036–1050 (2011)
5. Álvaro Jaramillo Duque, Castronuovo, E.D., Sánchez, I., Usaola, J.: Optimal operation of a pumped-storage hydro plant that compensates the imbalances of a wind power producer. Electric Power Systems Research 81(9), 1767 – 1777 (2011)
6. Mammen, D.L., Hogg, T.: A new look at the easy-hard-easy pattern of combinatorial search difficulty. JAIR 7, 47–66 (1997)
7. Pérez-Díaz, J.I., Wilhelmi, J.R., Arévalo, L.A.: Optimal short-term operation schedule of a hydropower plant in a competitive electricity market. Energy Conversion and Management 51(12), 2955 – 2966 (2010)
8. Zambelli, M., Huamani, I., Kadowaki, M., Soares, S., Ohishi, T.: Hydropower Scheduling in Large Scale Power Systems. INTECH Open Access Publisher (2012)
9. Zhang, X.M., Wang, L.p., Li, J.w., Zhang, Y.k.: Self-optimization simulation model of short-term cascaded hydroelectric system dispatching based on the daily load curve. Water Resources Management 27(15), 5045–5067 (2013)
10. Zhou, A., Qu, B.Y., Li, H., Zhao, S.Z., Suganthan, P.N., Zhang, Q.: Multiobjective evolutionary algorithms: A survey of the state of the art. Swarm and Evolutionary Computation 1(1), 32 – 49 (2011)
11. Zitzler, E., Laumanns, M., Thiele, L.: Spea2: Improving the strength pareto evolutionary algorithm. Eidgenössische Technische Hochschule Zürich (ETH), Institut für Technische Informatik und Kommunikationsnetze (TIK) (2001)