31792 Advanced Optimization and Game Theory for Power Systems

Peter Alexander Vistar Gade

August 12, 2021

Contents

1 Step 0: Test Case

The test case chosen for this assignment is the balancing market clearing problem [1]. The balancing market is cleared by the Transmission System Operator (TSO) and its purpose is to keep the production and consumption of electricity in balance at the lowest cost possible for the TSO. Imbalances generally arise from forecast inaccuracies of renewables and unexpected events such at outages at power plants. Imbalance also arises from noise in the equipment, transmission lines, etc. but this is negligible. The balancing market is close to real-time and is an energy market, i.e. participants get paid according to actual delivered and consumed energy as opposed to the day-ahead or reserve market where capacities and commitments are traded. After clearing of the balancing market, actors get paid according to either a one-price or two-price settlement scheme. In the one-price scheme, unknowing contributors to the restoration of the imbalance gets paid or pays the balancing price which arises from the market clearing, λ_B [€/MWh], whereas active participants to the restoration of imbalance also gets paid λ_B. In the two-price scheme, unknowing participants gets paid or pays the day-ahead market clearing price, λ_S, which removes the additional revenue arising from involuntarily helping the system restoration, and vice-versa with the loss.

In this assignment, the balancing market is considered with no connection to the day-ahead market and only for a one-hour time block. No busses or transmission constraints are taken into account. Also, price settlement is not discussed as both one-price and two-price settlement have the same balancing price. Thus, the clearing problem is formulated as shown in (1):

$$
\begin{aligned}
\min_{y_j^{\uparrow}, y_j^{\downarrow}} \quad & \sum_{j=1}^{N} \lambda_j^{\uparrow} y_j^{\uparrow} - \lambda_j^{\downarrow} y_j^{\downarrow} && \text{(1a)}\\
s.t \quad & \sum_{j=1}^{N} y_j^{\uparrow} - y_j^{\downarrow} = \Delta P: \quad \lambda_B, && \text{(1b)}\\
& 0 \leq y_j^{\uparrow} \leq P_j^{\uparrow}, \quad \forall j: \quad \mu_{1,j}, \mu_{2,j}, && \text{(1c)}\\
& 0 \leq y_j^{\downarrow} \leq P_j^{\downarrow}, \quad \forall j: \quad \mu_{3,j}, \mu_{4,j}, && \text{(1d)}
\end{aligned}
$$

The objective of (1) minimizes the need for balancing by choosing the best allocation of energy balancing quantities, $y_j^{\uparrow}$ and $y_j^{\downarrow}$ [MWh], arising from each generator's balancing offers that consists of prices, $\lambda_j^{\uparrow}$ and $\lambda_j^{\downarrow}$ [€/MWh], and capacities for up- and down regulation, $P_j^{\uparrow}$ and $P_j^{\downarrow}$ [MWh]. Equality constraint (1b) ensures that the net balancing energy bought by the TSO is exactly the imbalance, ΔP, in the network such that production and consumption are equal. The dual variable of the equality constraint is the balancing price after clearing, λ_B. The dual variables of the inequality constraints are $\mu_{1\ldots4,j}$. Constraints (1c, 1d) ensures that the generators' capacity for up- and down regulation are respected.

The problem in (1) scales with the number of generators in terms of number of variables and number of constraints as follows (where N is the number of generators):

- No. of variables: $2N$
- No. of constraints: $4N + 1$

Thus, in order to have a 1000 constraints, we have $N = 250$.

The problem in (1) is a simple linear program that can be solved with standard software and solvers. In this assignment, CVXPY [2] in Python has been chosen as the general framework to solve it with e.g. ECOS as the solver. Parameters in the problem, i.e. offer curves, imbalance, and capacities have been chosen randomly (with a seed for replication) albeit such that they make somewhat physical sense, i.e. the offers for up regulation are larger than the day-ahead price and vice versa. It is also assumed that the sum of available up or down regulation is always greater than the imbalance (otherwise, the problem wouldn't have a solution).

2 Step 1: Solving the problem picked in Step 0 in a deterministic manner (no uncertainty)

2.1 Solve the original (primal) problem in the programming language of your preference

As mentioned, the problem in (1) is solved in CVXPY in Python. The parameters are chosen as follows:

- No. of generators: $N = 250$
- Day-ahead price: $\lambda_S = 30$ [€/MWh]
- Imbalance: $\Delta P = 50$ [MWh]
- Down- and up regulation energy offers: drawn from a uniform distribution $U \sim \mathcal{U}(1, 50)$
- Down regulation offers: drawn from a uniform distribution $U \sim \mathcal{U}(1, \lambda_S - 1)$

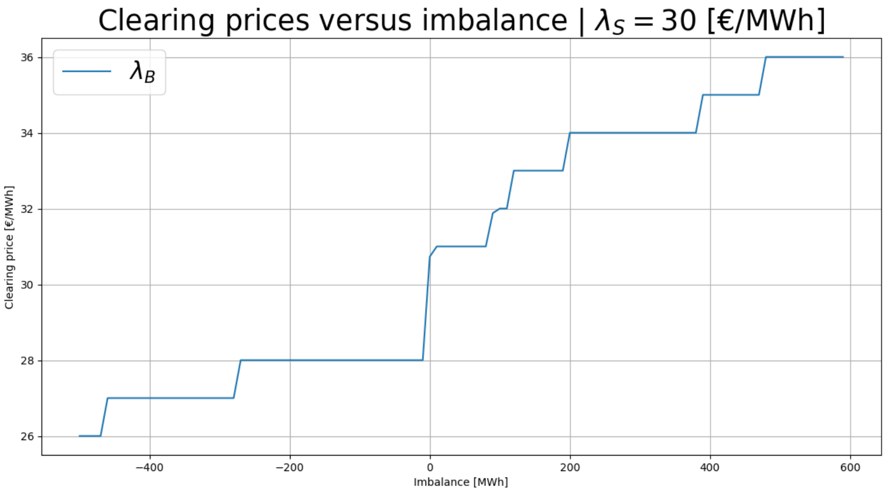


Figure 1: Simulation of problem (1) with different imbalances and resulting balancing prices using the conditions for the parameters outlined above.

- Down regulation offers: drawn from a uniform distribution $U \sim \mathcal{U}(\lambda_S + 1, 100)$

The code for solving problem (1) can be seen in the file, **step1.py**, in the uploaded attachment.

We can visualize how the balancing price changes as a function of the imbalance by solving problem (1) for a range of imbalances. This can be seen in Figure 1.

Here, it can be seen how the balancing price changes according to the sign of the imbalance. When the imbalance is negative, we have $\lambda_B \leq \lambda_S$ since generators need to buy back energy what they already sold at price λ_S. Likewise, when the imbalance is positive, i.e. production is lower than demand, $\lambda_S \leq \lambda_B$ since a generator needs to produce additional energy on top of what they already agreed to produce at price λ_S. Note that only the supply side (i.e. the generators) is considered but one would normally consider payments for both supply and demand and their deviations at the balancing stage from their respective day-ahead schedules.

The number of variables and constraints in this problem (with $N = 250$ generators) is 500 and 1001, respectively.

2.2 Derive the formulation of the dual optimization problem and solve it

The dual formulation of (1) requires computing the dual function and then setting up a max-min problem as described in [3] and solving it wrt. to the dual variables.

The dual function - or the Lagrangian - of problem (1) is:

$$\begin{aligned}\mathcal{L}(x, \lambda, \mu) &= f(x) + \lambda^T h(x) + \mu^T g(x) \\ &= \sum_{j=1}^{N} \lambda_j^{\uparrow} y_j^{\uparrow} - \lambda_j^{\downarrow} y_j^{\downarrow} + \lambda^B \left(\sum_{j=1}^{N} y_j^{\uparrow} - y_j^{\downarrow} - \Delta P \right) \\ &+ \sum_{j=1}^{N} \mu_{1,j} \left(y_j^{\uparrow} - P_j^{\uparrow} \right) - \sum_{j=1}^{N} \mu_{2,j} y_j^{\uparrow} \\ &+ \sum_{j=1}^{N} \mu_{3,j} \left(y_j^{\downarrow} - P_j^{\downarrow} \right) - \sum_{j=1}^{N} \mu_{4,j} y_j^{\downarrow}\end{aligned} \tag{2}$$

In (2), λ and μ is the clearing price and the slack variables, respectively, from problem (1). The dual problem can then be set up as:

$$\max_{\lambda_B,\mu} \quad \min_{y^\uparrow, y^\downarrow} \quad \mathcal{L}(x, \lambda, \mu) \tag{3}$$

Eq. (2) can be simplified by minimizing wrt. $\{y^\uparrow, y^\downarrow\}$ and only keeping the terms that are a part of the outer optimization in (3). Minimizing $\{y^\uparrow, y^\downarrow\}$ requires taking the partial derivative and setting it equal to zero. In this way, problem (3) can be rewritten to:

$$\max_{\lambda_B,\mu} \quad -\lambda_B \Delta P - \sum_{j=1}^{N} \mu_{1,j} P_j^\uparrow - \sum_{j=1}^{N} \mu_{3,j} P_j^\downarrow \tag{4a}$$

$$s.t \quad \lambda_j^\uparrow + \lambda_B + \mu_{1,j} - \mu_{2,j} = 0, \quad \forall j \tag{4b}$$

$$-\lambda_j^\downarrow - \lambda_B + \mu_{3,j} - \mu_{4,j} = 0, \quad \forall j \tag{4c}$$

Problem (4) is the dual problem of problem (1) which is called the primal problem. It maximizes the dual function at its minimum value (wrt. the primal variables, $\{y^\uparrow, y^\downarrow)$ wrt. to the dual variables, $\{\lambda_B, \mu\}$. Thus, problem (4) is concave in nature; it pushes the minimum value that the dual function can take as high as possible through the dual variables. One can think of the dual function as the unconstrained version of the primal problem in (1) where the dual problem in (4) penalizes violations of the constraints through $\{\lambda_B, \mu\}$ since small values of $\{\lambda_B, \mu\}$ lead to a higher objective value in the dual problem.

The dual problem was solved in the same manner as the primal problem using CVXPY in Python. The number of variables equals the number of constraints in the primal problem, i.e. 1001, and the number of constraints equals the number of variables in the primal problem, i.e. 500.

In Figure 2, it can be seen how the balance price evolves with varying imbalances. The balancing prices obtained from the dual problem from the variable vector are exactly the same ones as the balancing prices obtained from the primal problem in the dual to the equality constraint (the solver automatically calculates the duals).

2.3 Derive the formulation of KKT conditions and solve it

The KKTs represents sufficient conditions for optimality. The KKTs are a system of equations given by:

$$\frac{\partial \mathcal{L}(y, \lambda, \mu)}{\partial y} = 0 \tag{5a}$$

$$h(y) = 0 \tag{5b}$$

$$0 \leq -g(y) \quad \perp \quad \mu \geq 0 \tag{5c}$$

$$\lambda \in \text{free} \tag{5d}$$

Inserting into eq. (5) from the primal problem in (1) yields the KKTs:

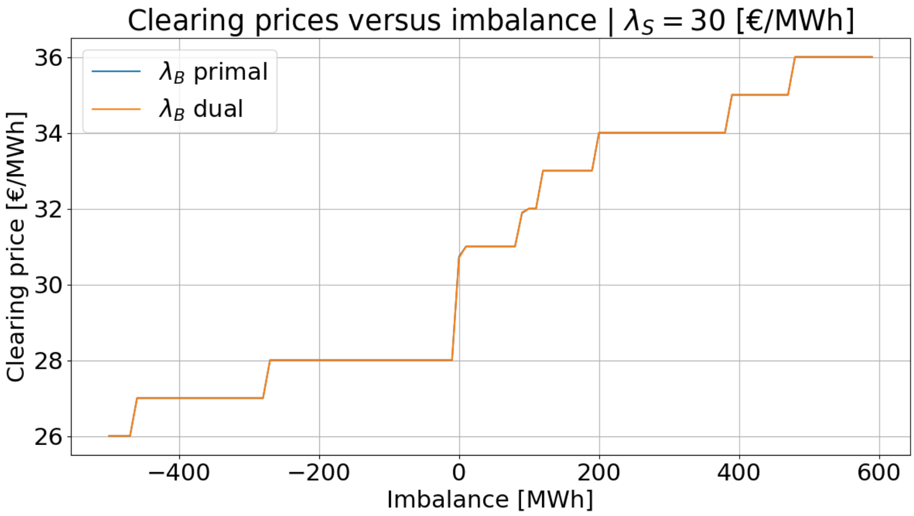


Figure 2: Simulation of problem (4) with different imbalances as in the primal problem. A comparison with the clearing price obtained from both problems is shown. As can be seen, they are exactly equal as they should.

$$\lambda_j^{\uparrow} + \lambda_B + \mu_{1,j} - \mu_{2,j} = 0, \quad \forall j \tag{6a}$$

$$-\lambda_j^{\downarrow} - \lambda_B + \mu_{3,j} - \mu_{4,j} = 0, \quad \forall j \tag{6b}$$

$$\sum_{j=1}^{N} \left(y_j^{\uparrow} - y_j^{\downarrow} \right) - \Delta P = 0 \tag{6c}$$

$$\mu_{1,j} \geq 0, \quad \forall j \quad \perp \quad y_j^{\uparrow} - P_j^{\uparrow} \leq 0 \tag{6d}$$

$$\mu_{2,j} \geq 0, \quad \forall j \quad \perp \quad -y_j^{\uparrow} \leq 0 \tag{6e}$$

$$\mu_{3,j} \geq 0, \quad \forall j \quad \perp \quad y_j^{\downarrow} - P_j^{\downarrow} \leq 0 \tag{6f}$$

$$\mu_{4,j} \geq 0, \quad \forall j \quad \perp \quad -y_j^{\downarrow} \leq 0 \tag{6g}$$

Equations (6d)-(6g) are complementarity conditions, which means that the two sub-conditions in such a condition are mutually exclusive wrt. being strict inequalities. It arises from the fact that $g(y)\mu = 0$. The complementarity conditions are non-linear which makes eqs. (6) harder to solve than a simple linear program as in (1).

For this, the JuMP package from Julia has beed used with the Ipopt solver by reformulating (6) into a minimization problem of a constant with constraints given by the equations. Figure 3 shows the clearing price from solving the KKTs for different imbalances and they are matching the clearing prices obtained from the primal and dual problems. Given $N = 250$ generators, there are $2N + 4N + 1 = 1501$ variables and $2N + 12N + 1 = 3501$ constraints in (6).

The code for solving the KKTs can be seen **kkt.jl** in the uploaded attachment.

3 Step 2: Solving the problem under uncertainty

For this step, the problem in (1) has been modified to properly account for uncertainty such that stochastic programming can be applied. Specifically, in addition to the real-time balance market clearing, the reservation market (which usually occurs about 24 hours beforehand) is also included. The reservation market is a capacity market with no energy costs but only

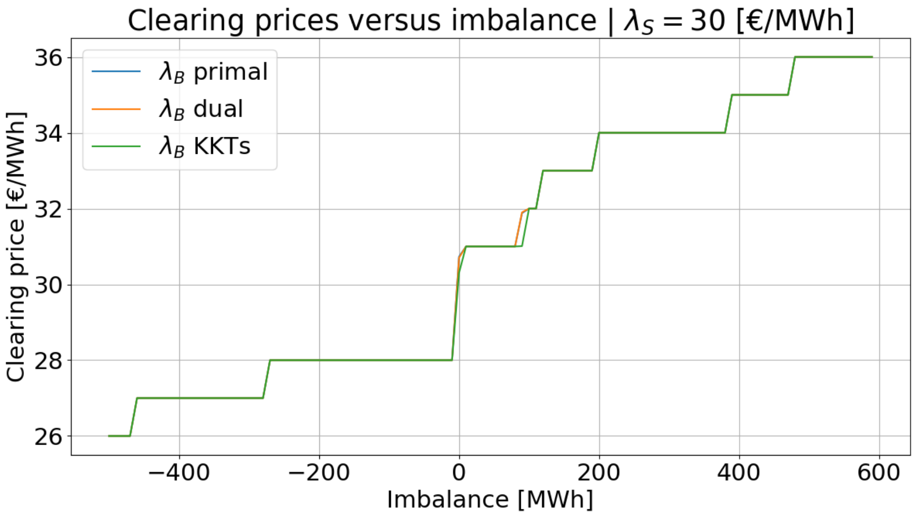


Figure 3: Simulation of the KKTs in (6) with different imbalances as in the primal and dual problem. A comparison with the clearing price obtained from both problems is shown. As can be seen, they are exactly equal as they should.

reservations with the possibility to be activated at the balancing stage. The TSO also clears that market. At the time of clearing of the reservation market, the actual real-time imbalance the next day, ΔP from (1), is uncertain. This gives rise to the following formulation:

$$\min_{r_j^{\uparrow}, r_j^{\downarrow}, y_{j,\omega}^{\uparrow}, y_{j,\omega}^{\downarrow}} \quad \underbrace{\sum_{j=1}^{N} c_j^{\uparrow} r_j^{\uparrow} + c_j^{\downarrow} r_j^{\downarrow}}_{\text{First stage}} + \underbrace{\sum_{\omega=1}^{M} \sum_{j=1}^{N} \left(\lambda_j^{\uparrow} y_{j,\omega}^{\uparrow} - \lambda_j^{\downarrow} y_{j,\omega}^{\downarrow} \right) \cdot p_\omega}_{\text{Second stage}} + \underbrace{\sum_{\omega=1}^{M} W_\omega^{shed} C^{shed} \cdot p_\omega}_{\text{Cost of shedding}} \tag{7a}$$

$$s.t \quad 0 \leq r_j^{\uparrow} \leq R_{j,max}^{\uparrow}, \quad \forall j, \tag{7b}$$

$$0 \leq r_j^{\downarrow} \leq R_{j,max}^{\downarrow}, \quad \forall j, \tag{7c}$$

$$\sum_{j=1}^{N} \left(y_{j,\omega}^{\uparrow} - y_{j,\omega}^{\downarrow} \right) + W_\omega^{shed} - W_\omega^{spill} = \Delta P_\omega, \quad \forall \omega \in \Omega \tag{7d}$$

$$0 \leq y_{j,\omega}^{\uparrow} \leq r_j^{\uparrow}, \quad \forall j, \forall \omega \in \Omega, \tag{7e}$$

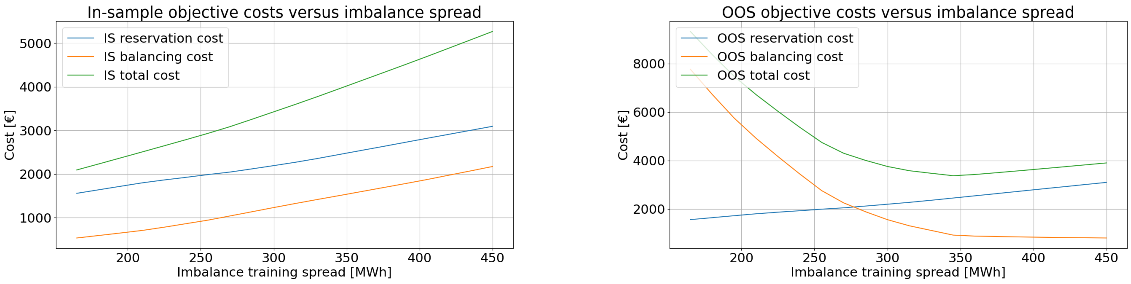
$$0 \leq y_{j,\omega}^{\downarrow} \leq r_j^{\downarrow}, \quad \forall j, \forall \omega \in \Omega, \tag{7f}$$

$$0 \leq W_\omega^{shed}, \quad \forall \omega \in \Omega, \tag{7g}$$

$$0 \leq W_\omega^{spill}, \quad \forall \omega \in \Omega \tag{7h}$$

Problem (7) is then a two-stage stochastic program where:

- **The first stage** clears the reservation market given generator offers $c_j^{\uparrow}$, $c_j^{\downarrow}$ and maximum balancing capacities $R_{j,max}^{\uparrow}$, $R_{j,max}^{\downarrow}$ resulting in reserved capacities, $r_j^{\uparrow}$, $r_j^{\downarrow}$, for each generator j.
- **The second stage** minimizes the expected cost under all scenarios $\forall \omega \in \Omega$ of balancing and shedding. Constraint (7d) ensures power balance under each scenario where it might be necessary for some demand to curtail or shed load at a great cost and/or some generators to spill generation at no cost. The two stages are coupled through constraints (7e)-(7f) which ensures that needed balancing quantities, $y_{j,\omega}^{\uparrow}$ and $y_{j,\omega}^{\downarrow}$, are limited by the reservation quantities.



(a) In-sample cost versus imbalance spread. The cost is decomposed into the reservation cost and the balancing cost.

(b) Out-of-sample cost versus imbalance spread. The cost is decomposed into the reservation cost and the balancing cost.

Figure 4

Importantly, the reservations are determined by taking future uncertainty into account. For example, if one considers only scenarios with small imbalances it is entirely possible that the reservation capacities can not cover large imbalances at the real-time stage.

The number of variables and constraints in problem (7) is $2N + 2NM + 2M$ and $4N + M + 4NM + 2M$, respectively. For $N = 10$ generators and $M = 100$ scenarios, this yields $20 + 2000 + 200 = 2220$ variables and $40 + 100 + 4000 + 200 = 4340$ constraints.

3.0.1 Analysis 1

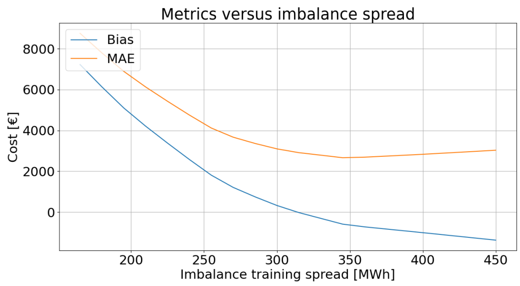
The first analysis investigates the impact of the scenarios chosen in the training phase:

- Create $M = 80$ scenarios of imbalances, ΔP_ω, from a normal distribution with mean $\mu = 0$ MWh and standard deviation $\sigma = 75$ MWh to be used as a test set.
- Repeat he following steps for each $i = 0...19$.
- Create 20 scenarios as training samples with imbalances specified with lower bound, $\mu - 3\sigma + 0.10\sigma \cdot i$, and upper bound, $\mu + 3\sigma - 0.10\sigma \cdot i$. Denote the spread as δ.
- Solve problem (7) using the training set and obtain optimal reservation quantities, $r_j^{\uparrow,\star}$, $r_j^{\downarrow,\star}$. The problem is solved once for the whole training set.
- Solve problem (7) using the test set with *fixed* reservation quantities, $r_j^{\uparrow,\star}$, $r_j^{\downarrow,\star}$, obtained in the training phase and calculate the objective function cost. Importantly, each test sample, i.e. each individual imbalance ΔP_ω is solved individually. Thus, problem (7) is solved 80 times.

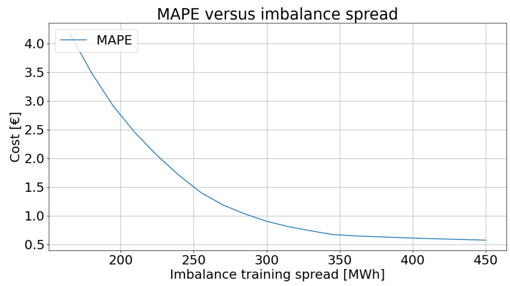
Figure 4 shows the in-sample and oos-sample costs for different imbalance spreads that the model is trained on (and thus subsequent reservation quantities). In general, it gets more and more expensive to account for a wider range of scenarios (a higher spread), as seen in Figure 4a for the in-sample cost whereas the oos-cost is minimized at a spread of about 330 MWh. This is interesting; the oos reservation cost gets to big compared to the balancing cost for spreads above 330 MWh as seen in Figure 4b.

The bias[1] is also close to 0 around 330 MWh and the same is the mean absolute error (MAE) as seen in Figure 5a. The percentage difference between the in-sample and oos objective costs are also not getting much better for $\delta > 330$ MWh as seen in Figure 5b.

[1]The bias should be 0 for any model, otherwise there will be a consistent bias of either under- or overestimation.



(a) Cost metrics versus imbalance spread. The bias represents the difference between the in-sample and out-of-sample objective function value, and the mean absolute error represents the absolute difference.



(b) Mean absolute percentage error (MAPE) of the in-sample and out-of-sample cost functions versus imbalance spread. The MAPE represents percentage deviation of the in-sample cost.

Figure 5

Concluding, it seems that there is a trade-off when choosing the scenarios: there is clearly a need to include large imbalances but it is not necessarily optimal to be completely conservative. This analysis of course also depends on the test set which in this case is centered around a $N(0, 75)$ distribution so there is probably not too may cases of very large imbalances (the training set was created with up to ± 3 standard deviations). However, it is better to be conservative than not, i.e. it is better to choose a conservative spread rather than a narrow spread.

3.0.2 Analysis 2

For the second analysis, an additional term to the problem in (7) is considered called Conditional Value-at-Risk (CVaR) such that the problem in (7) is now [1]:

$$\min_{x} \quad (1-k)\,\mathbb{E}_{\omega}\left[f(x,\omega)\right] + k \cdot \mathrm{CVaR}_{1-\alpha}(x) \tag{8a}$$

$$s.t \quad \mathrm{CVaR}_{1-\alpha}(x) = \zeta + \frac{1}{1-\alpha}\sum_{\omega=1}^{M} p_{\omega}\eta_{\omega} \tag{8b}$$

$$f(x,\omega) - \zeta \leq \eta_{\omega}, \quad \forall \omega \in \Omega, \tag{8c}$$

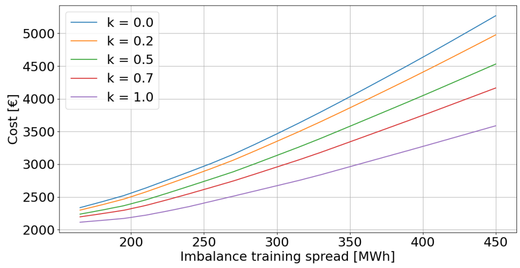
$$\eta_{\omega} \geq 0, \quad \forall \omega \in \Omega, \tag{8d}$$

$$Ax \leq b, \tag{8e}$$

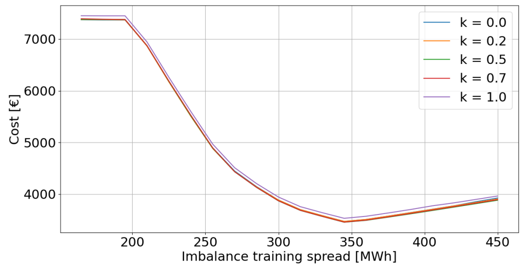
$$x \geq 0 \tag{8f}$$

Here, $\mathbb{E}_{\omega}\left[f(x,\omega)\right]$ and constraints (8e)-(8f) are the objective function and constraints of problem (8), respectively. The new variables introduced in problem (8) are the Value-at-Risk (VaR), ζ, CVaR at the $1-\alpha$ quantile, and auxillary variables η_{ω}, $\forall \omega \in \Omega$ that represents the cost (or profit) distance from a given solution, $f(x,\omega)$, to ζ. This distance is then minimized as much as possible under each scenario such that the CVaR is minimized along with the rest of the objective function. The constant k represents a trade-off between the reservation and balancing cost versus the risk associated with the those costs. The term $\frac{1}{1-\alpha}$ represents the importance the quantile, i.e. more risk averse solutions will weigh the CVaR correspondingly higher in the objective function.

In short, the CVaR is the expected cost of the $1-\alpha$ percent worst scenarios in Ω. Thus, it takes the area or the tail of the cost distribution function into account unlike the VaR. By including it in the optimization problem with $k > 0$, the solution obtained will be more robust



(a) In-sample total cost versus imbalance spread for different k.



(b) Out-of-sample cost versus imbalance spread for different k.

Figure 6

towards scenarios in the (fat) tail of the cost distribution and can thus represent a form of hedge towards risk. The idea is that this will perform well out-of-sample since high cost scenarios are weighted.

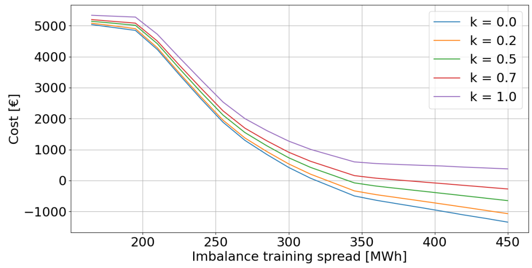
The analysis carried out with the formulation of the CVaR in (8) is specified as the same as analysis 1 for different k:

- Create $M = 80$ scenarios of imbalances, ΔP_ω, from a normal distribution with mean $\mu = 0$ MWh and standard deviation $\sigma = 75$ MWh to be used as a test set.
- Repeat he following steps for each $i = 0...19$.
- Create 20 scenarios as training samples with imbalances specified with lower bound, $\mu - 3\sigma + 0.10\sigma \cdot i$, and upper bound, $\mu + 3\sigma - 0.10\sigma \cdot i$. Denote the spread as δ.
- Set $\alpha = 0.95$.
- Repeat the following for each k in $\{0, 0.20, 0.50, 0.70, 1\}$.
- Solve problem (8) using the training set and obtain optimal reservation quantities, $r_j^{\uparrow,\star}$, $r_j^{\downarrow,\star}$.
- Solve problem (8) with $k = 0$ using the testing set with *fixed* reservation quantities, $r_j^{\uparrow,\star}$, $r_j^{\downarrow,\star}$, obtained in the training phase calculate the objective function cost. Solve it once per test sample.

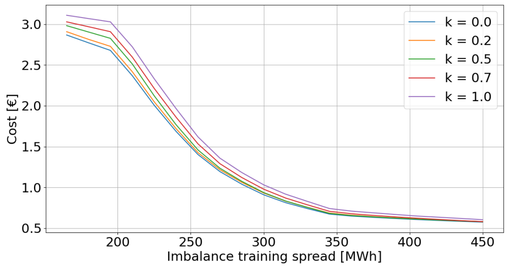
The analysis will then investigate the implications of expected market cost versus CVaR cost as the importance of CVaR varies when training the in-sample model. In Figure 6a, it can be seen how the in-sample cost decreases by increasing the importance of the CVaR. This is probably due to the fact that it will get less costly in total when taking CVaR into account such that the overall cost due to reservation, balancing cost, and CVaR is cheaper. For the OOS cost in Figure 6b, the cost is computed just as in problem 7 (since CVaR doesn't make sense to include when testing a single realization OOS). Here, we see no significant differences between including CVaR but these differences will probably be more pronounced for more fat-tailed distributions.

Figures 7a and 7b shows the bias and MAPE and here it seems that it's better to include the CVaR since the bias is closer to 0 for $k = 0.2$, $k = 0.5$, and $k = 0.7$ for $\delta > 330$ MWh.

Concluding, the CVaR formulation seems to slightly improve the in-sample cost (when measured as the objective cost in problem (8)) but there are no significant changes to the out-of-sample cost (when measured as in problem (7)).

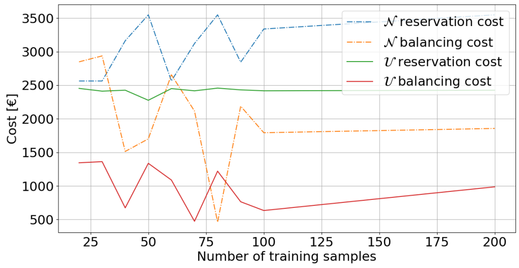


(a) Bias between IS and OOS cost versus imbalance spread for different k.

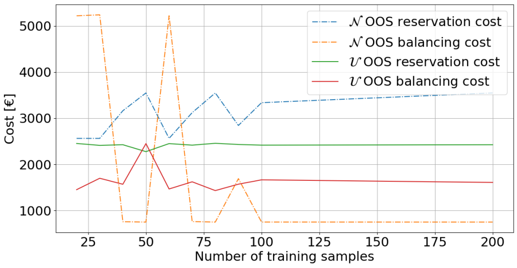


(b) MAPE between IS and OOS cost versus imbalance spread for different k.

Figure 7



(a) IS costs when training on a $\mathcal{N}(0, 150)$ and a $\mathcal{U}(-150, 150)$ training set, respectively.



(b) OOS costs when training on a $\mathcal{N}(0, 150)$ and a $\mathcal{U}(-150, 150)$ training set, respectively.

Figure 8

3.0.3 Analysis 3

In this analysis, the training sample distribution is investigated by comparing the normal- and uniform distributions when drawing training samples. The analysis follow the same procedure as the previous analyses except with $\delta = 300$ MWh and $k = 0$. Figure 8 shows the IS and OOS cost breakdown for the two distributions as a function of the number of training samples. Both of them seems to stabilize when increasing the number of training samples. Interestingly, the Gaussian distribution leads to higher reservation costs compared to the uniform distribution *and* higher balancing costs in the training phase but *lower* balancing costs in the testing phase. Thus, it seems that the Gaussian distribution is better suited for drawing random samples to the training set than the uniform one wrt. performing well OOS in the balancing stage. There is a natural explanation: since the uniform distribution is bounded by $\Delta P \in \{-150, 150\}$, and the $\mathcal{N}(0, 150)$ is not, it means that there is a much wider range of imbalances in the training phase for the Gaussian distribution.

However, one would also expect that it would still be centered around 0 and thus assigning the biggest weight to those as opposed to the uniform distribution which would yield training samples equally spread over $\Delta P \in \{-150, 150\}$. It seems the formulation in (7) is robust in that sense, i.e. it matters a lot for the OOS performance that outliers are present in the training set, even though not frequently present.

4 Step 3: Solving a bilevel program

In this step, a slightly different aspect of the market clearing problem in the previous steps is considered. Specifically, two new variables will be introduced which are the reserve requirements for up regulation and down regulation, respectively. In reality, the TSO will somehow estimate the need for reserve requirements *before* clearing the reservation market and subsequent markets. The choice of reservation requirements is therefore quite crucial for the reservation cost and and balancing cost. In this step, the goal is to optimally choose reservation requirements and reserve schedules at once while taking into account the real-time balancing markets and its realizations of actual imbalances; similar to the previous steps.

Three mathematical models will be compared: determination of reserve requirements through a bilevel program, an ideal stochastic program, and a sequentially cleared market, respectively. The ideal stochastic program is identical to the program in (7) except with reserve requirements as two new variables:

$$\begin{aligned}
\min_{\Phi \in \{(7)\}, D^{\uparrow}, D_{\downarrow}} \quad & (7) && \text{(9a)}\\
s.t \quad & (7b) - (7h) && \text{(9b)}\\
& 0 \leq D^{\uparrow}, && \text{(9c)}\\
& 0 \leq D^{\downarrow}, && \text{(9d)}
\end{aligned}$$

Here, $\Phi \in \{(7)\}$ is the set of decision variables in problem (7), i.e. reserve schedules, real-time dispatches with shedding and spilling. Problem (9) will find an *ideal* solution and reserve requirements because it is a joint optimization over reservation *and* real-time scenarios. This means that the resulting reserve schedule and reserve requirements are optimal in expectation of the uncertainty of the actual imbalance in the real-time market, but it violates some important properties such as individual rationality (generators can face negative payments some times) and the merit order effect (no one is guaranteed reservation even though ones offer price might be lower than all the others). For these reasons, stochastic market clearing is not used in Europe. However, it serves as an ideal benchmark for other problems.

As mentioned, in reality, the reserve schedule is cleared without taking the real-time market and its uncertainty into account. By doing that, the clearing problem adheres to the individual rationality for each participant (i.e. generator in this case) and it respects the merit order effect. In the following, a bilevel program is presented that tries to approximate the ideal stochastic problem in (9) while respecting the sequential nature of the decision making and still taking uncertainty of the real-time market into account:

$$\begin{aligned}
\min_{\Phi \in \{(7)\} \setminus \{r_j^{\uparrow}, r_j^{\downarrow}\}, D^{\uparrow}, D_{\downarrow}} \quad & (7) && \text{(10a)}\\
s.t \quad & (7b) - (7h) && \text{(10b)}\\
& 0 \leq D^{\uparrow}, && \text{(10c)}\\
& 0 \leq D^{\downarrow}, && \text{(10d)}\\
& \{r_j^{\uparrow}, r_j^{\downarrow}\} \in \min_{\{r_j^{\uparrow}, r_j^{\downarrow}\}} \sum_{j=1}^{N} c_j^{\uparrow} r_j^{\uparrow} + c_j^{\downarrow} r_j^{\downarrow} && \text{(10e)}\\
& \qquad s.t. \quad 0 \leq r_j^{\uparrow} \leq R_{j,max}^{\uparrow}, \quad \forall j, && \text{(10f)}\\
& \qquad \qquad 0 \leq r_j^{\uparrow} \leq R_{j,max}^{\uparrow}, \quad \forall j, && \text{(10g)}\\
& \qquad \qquad \sum_{j}^{N} r_j^{\uparrow} = D^{\uparrow}, && \text{(10h)}\\
& \qquad \qquad \sum_{j}^{N} r_j^{\downarrow} = D^{\downarrow}, && \text{(10i)}
\end{aligned}$$

The bilevel problem in (10) has a lower level problem that optimizes the reserve schedule *given* some fixed reserve requirements, $\{D^{\uparrow}, D^{\downarrow}\}$, from the upper problem. The upper problem then takes into account the real-time imbalance uncertainty when choosing $\{D^{\uparrow}, D^{\downarrow}\}$ but, crucially, the reserve schedule in the lower problem adhere to the current reality by choosing the optimal schedule without knowledge about the future (which is implicit through $\{D^{\uparrow}, D^{\downarrow}\}$). In short, problem (10) finds the optimal reserve requirements, $\{D^{\uparrow}, D^{\downarrow}\}$ that leads to the most optimal reserve schedule while ensuring non-anticipativity through a feedback loop with the lower level problem's resulting reserve schedule.

To solve (10), the lower level problem is replaced with its KKTs:

$$
\begin{aligned}
\min_{\Phi \in \{(7)\}, D^{\uparrow}, D_{\downarrow}} \quad & (7) && \text{(11a)}\\
s.t \quad & (7\text{b}) - (7\text{h}) && \text{(11b)}\\
& 0 \leq D^{\uparrow}, && \text{(11c)}\\
& 0 \leq D^{\downarrow}, && \text{(11d)}\\
& c_j^{\uparrow} - \mu_{1,j} + \mu_{3,j} + \lambda^{(1)} = 0, \quad \forall j, && \text{(11e)}\\
& c_j^{\downarrow} - \mu_{2,j} + \mu_{4,j} + \lambda^{(2)} = 0, \quad \forall j, && \text{(11f)}\\
& \sum_{j}^{N} r_j^{\uparrow} = D^{\uparrow}, && \text{(11g)}\\
& \sum_{j}^{N} r_j^{\downarrow} = D^{\downarrow}, && \text{(11h)}\\
& 0 \leq r_j^{\uparrow}, \quad \forall j, && \text{(11i)}\\
& 0 \leq r_j^{\downarrow}, \quad \forall j, && \text{(11j)}\\
& r_j^{\uparrow} \leq r_{j,max}^{\uparrow}, \quad \forall j, && \text{(11k)}\\
& r_j^{\downarrow} \leq r_{j,max}^{\downarrow}, \quad \forall j, && \text{(11l)}\\
& 0 \leq \mu_{i,j}, \quad \forall i, \quad \forall j, && \text{(11m)}\\
& 0 \leq r_{j,max}^{\uparrow} - r_j^{\uparrow}, \quad \forall j, && \text{(11n)}\\
& 0 \leq r_{j,max}^{\downarrow} - r_j^{\downarrow}, \quad \forall j, && \text{(11o)}\\
& r_j^{\uparrow} \leq \theta_1 \cdot M, \quad \forall j, && \text{(11p)}\\
& \mu_{1,j} \leq (1 - \theta_1) \cdot M, \quad \forall j, && \text{(11q)}\\
& r_j^{\downarrow} \leq \theta_2 \cdot M, \quad \forall j, && \text{(11r)}\\
& \mu_{2,j} \leq (1 - \theta_2) \cdot M, \quad \forall j, && \text{(11s)}\\
& r_{j,max}^{\uparrow} - r_j^{\uparrow} \leq \theta_3 \cdot M, \quad \forall j, && \text{(11t)}\\
& \mu_{3,j} \leq (1 - \theta_3) \cdot M, \quad \forall j, && \text{(11u)}\\
& r_{j,max}^{\downarrow} - r_j^{\downarrow} \leq \theta_4 \cdot M, \quad \forall j, && \text{(11v)}\\
& \mu_{4,j} \leq (1 - \theta_4) \cdot M, \quad \forall j, && \text{(11w)}\\
& \theta_i \in \{0, 1\}, \quad \forall i && \text{(11x)}
\end{aligned}
$$

Problem (11) contains many constraints due to replacing the lower problem with its KTTs. Its complementarity constraints are linearized using the Big-M approach where $M = 10^3$ is sufficient to solve it. The variables, θ_i, are integer variables which are necessary to include due to the complementarity constraints from the KKTs.

Problem (11) has been solved with the same specifications as in the previous steps, i.e. with 100 scenarios of actual real-time imbalances (where 80 are for testing and 20 are for train) and 10 generators. Remember, the 80 testing samples are used in such a way that only the real-time problem is solved (once per sample) with a fixed reserve schedule arising from (9) and (11). The training and testing sets both span imbalances from -150 MWh to 150 MWh.

The analysis is focused on a comparison between problem (11) and (9), i.e. the bilevel model and the ideal stochastic model. However, solving (9) in a sequential manner for comparison has also been implemented. The models are solved in Python using CVXPY with CBC as the solver for the MILP in (11).

Table 1 compares the resulting cost components and test metrics between the three models. It can be seen that the ideal model has lower costs in-sample and out-of-sample which is to be expected. The correspondence between in-sample and out-of-sample is also better for the ideal model as seen in the lower error measures, i.e. MAE and MAPE. Interestingly, this main difference between the models seem to be mostly due to the reserve schedule which has a lower cost for the ideal model. Besides, the downwards reserve requirement is exactly the same for the two models while the ideal model does not need to purchase as much upward reserve as the bilevel model.

The sequential model is the worst of the three out-of-sample. Here, reserve requirements are also made very simple with only 30 MWh. The benefit of this is obviously very low day-ahead costs and very low OOS costs when the realization is in $\Delta P \in$ [-30:30] MWh.

Table 1: Comparison between the bilevel model (11), the ideal stochastic model (9), and the sequential model.

	Bilevel	Ideal	Sequential
IS day ahead cost	1410	1304	180
IS real time cost	3187	2998	0
IS total cost	4598	4302	180
OOS real time cost	3175	2960	28892
OOS total cost	4585	4264	29072
Bias	-12	-37	28892
MAE	1790	1555	28892
std	2265	1998	37933
MAPE	0.39	0.36	160.51
Reservation requirements	[174.0, 102.63]	[150.0, 102.63]	[30.0, 30.0]

Table 2 shows the reserve schedule for the three models (and also the generator offers). Interestingly, it can be seen exactly how the ideal stochastic model does not always choose the cheapest generators. This is due to the anticipation of the uncertainty in the real-time market and it is therefore sometimes more optimal to allocate reserve to a more expensive generator if it has a larger flexibility and capacity. The ideal model also does not need to allocate as much reserve as the bilevel model in total due to the smart way it allocates in anticipation of the future flexibility. This is seen in Table 1 as well. The sequential model just chooses the cheapest possible (like the bilevel) as expected.

Lastly, Figure 9 shows the OOS cost for the 80 testing samples for both models and the sequential model. Interestingly, the difference is fairly constant which highlights the fact that the main difference between the model is in the reserve schedule and the reservation cost (which has already been stated). The bilevel model is thus good enough at adapting to adverse real-time imbalances albeit at a slightly higher reservation cost.

Table 2: Generator comparison between the bilevel model (11), the ideal stochastic model (9), and the sequential model as well as generator offers.

	Bilevel	Ideal	Sequential	MWh offers	price offers
G0-up	18	18	0	18	5
G1-up	30	9	0	30	7
G2-up	21	21	0	21	9
G3-up	0	0	0	3	9
G4-up	6	6	0	6	5
G5-up	38	0	0	38	8
G6-up	13	0	0	13	8
G7-up	45	45	30	45	3
G8-up	3	3	0	3	8
G9-up	0	48	0	48	9
G0-down	0	0	0	9	5
G1-down	0	0	0	2	9
G2-down	0	0	0	18	6
G3-down	0	17	0	49	5
G4-down	17	0	0	36	4
G5-down	0	0	0	28	8
G6-down	0	0	0	41	7
G7-down	37	37	30	37	3
G8-down	49	49	0	49	3
G9-down	0	0	0	26	4

Here, it can be clearly seen how the sequential model is really bad OOS for large, positive realizations. However, it should be noted that it works fine for large, negative realizations which is due to the model choosing spilling instead of down regulation since spilling is free in problems (9) and (11). In reality, one would prefer to purchase the down regulation instead due to the payment and settlements process that happens afterwards but this is not included here.

Figure 10 shows the OOS cost for deterministic reserve requirements of 150 MWh (up and down) instead of 30 MWh. Here, the sequential model is still the worst, suggesting that the bilevel model should be used instead to determine reserve requirements since it also preservers the merit order effect and individual rationality.

5 Step 4: Decomposition

In this step, the goal is to solve the usual stochastic market clearing problem in (7) i a decomposed manner. Usually, decomposition is used to make intractable problems tractable or make hard problems faster to solve. The problem in (7) is easy so throughout this step, a comparison will be made to the ideal stochastic solution which is just to solve (7) as a normal LP.

The decomposition technique used here is the consensus ADMM where all the agents work together to agree on a common variable through iterative updates. ADMM is one type of (Augmented) Lagrangian Relaxation where complicating constraints are relaxed such that the problem is decomposed. In consensus ADMM, there is a single variable that all agents need to agree on through a shared (complicating) constraint. In problem (7), this is the reserve schedule, $\{r^{\uparrow}, r^{\downarrow}\}$, and the agents are all the scenarios, $\forall \omega \in \Omega$. Specifically, an equivalent version of (7) contains non-anticipativity constraints for the reserve schedules, $r_{\omega} = \omega'$, $\forall \omega$, $\forall \omega'$. This constraint is then relaxed in consensus ADMM. Thus, the problem can be decomposed by scenarios where each scenario will have a subproblem that is easily solvable [4]:

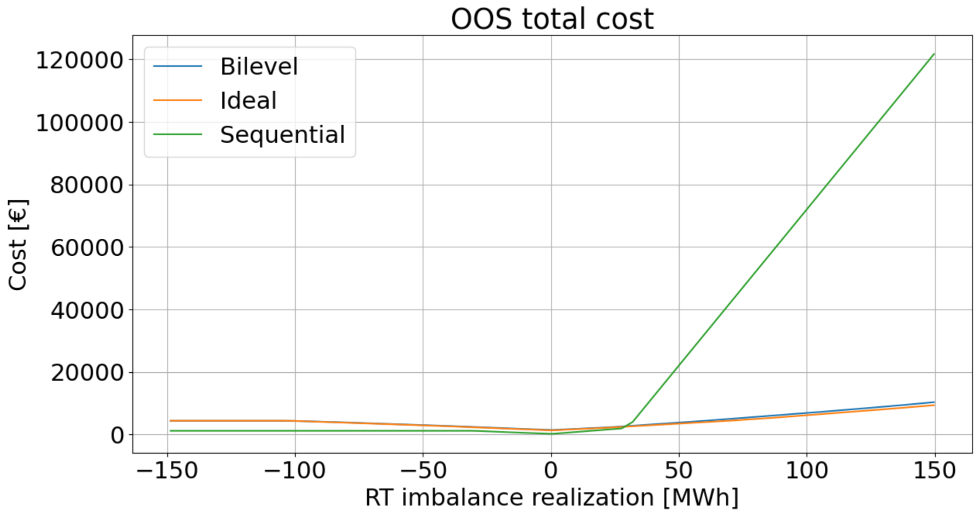


Figure 9: OOS cost versus realizations. Here, the reserve requirements for the sequential model are set to 30 MWh for both up and down regulation.

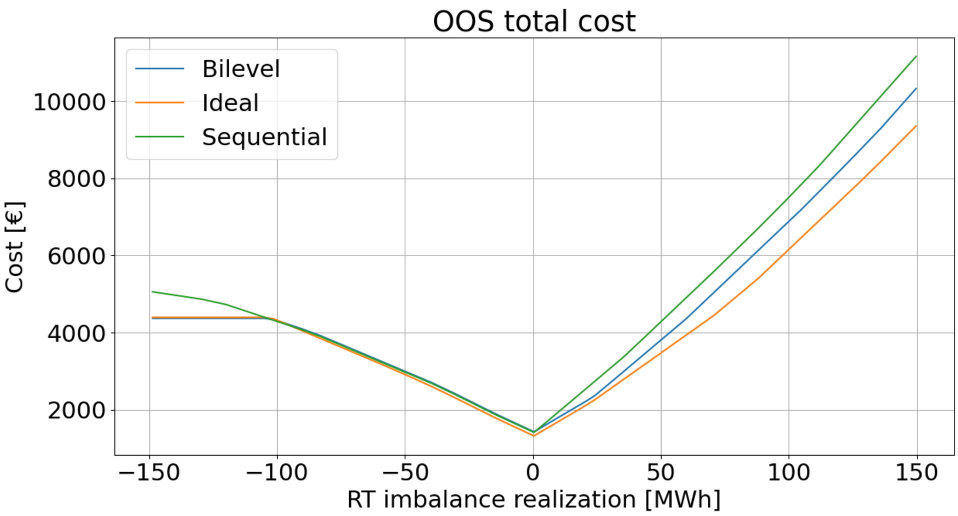


Figure 10: OOS cost versus realizations. Here, the reserve requirements for the sequential model are set to 150 MWh for both up and down regulation.

$$\{r^{\uparrow}, r^{\downarrow}\}\text{-update} = \begin{cases} \min\limits_{r_j^{\uparrow,\nu}, r_j^{\downarrow,\nu}, y_{j,\omega}^{\uparrow,\nu}, y_{j,\omega}^{\downarrow,\nu}} & (7) \cdot \pi + \sum_j^N \bar{\lambda}_{j,\omega}^{\nu-1} r_{j,\omega}^{\nu} + \sum_j^N \frac{\gamma}{2} \left\| r_{j,\omega}^{\nu} - \bar{r}_j^{\nu-1} \right\|^2, \quad \forall \omega,\ r_{j,\omega}^{\nu} \in \{r_{j,\omega}^{\uparrow,\nu}, r_{j,\omega}^{\downarrow,\nu}\} \\ \text{s.t.} \quad (7b) - (7h) & \end{cases} \tag{12a}$$

$$\bar{\lambda}\text{-update} = \left\{ \bar{\lambda}_{j,\omega}^{\nu} \leftarrow \bar{\lambda}_{j,\omega}^{\nu} + \gamma \left(r_{j,\omega}^{\nu} - \bar{r}_j^{\nu} \right), \quad \forall j,\ \forall \omega,\ r_{j,\omega}^{\nu} \in \{r_{j,\omega}^{\uparrow,\nu}, r_{j,\omega}^{\downarrow,\nu}\} \right. \tag{12b}$$

Eq. (12a) then consists of $\forall \omega \in \Omega$ subproblems at iteration ν. For each subproblem, the reserve schedule is solved for that subproblem, i.e. scenario, only. This is done for all subproblems and then two updates in 12a and (12b) are applied. For every iteration, each scenarios's reserve schedule will converge to the global mean reserve schedule over all scenarios, i.e. $\bar{r}_j$. Notice that (12a) contains quadratic terms in the objective function; it's still convex but harder to solve than a simple LP.

Solving problem (12), it can be seen how the reserve cost develops per iteration in Figure 11. Here, the error decreases quickly for only a few iterations and the day-ahead cost also stabilized although more iterations could be beneficial. There is probably ample room for refining and improving the optimization procedure; for example by varying the step size parameter, γ, throughout the iterations.

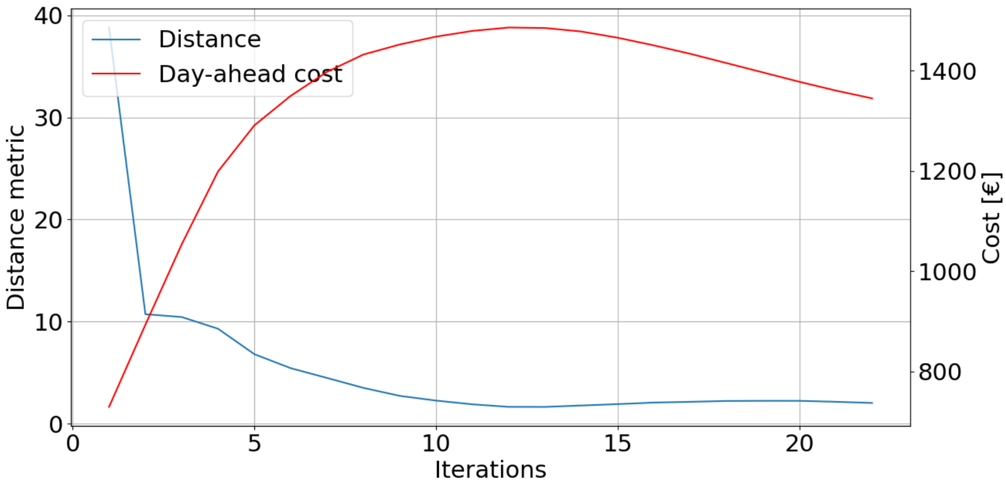


Figure 11: Distance metric and day-ahead cost per iteration in problem 12. The distance metric is given by $\sqrt{\sum_j^N \left(\bar{r}_j^{\nu} - \bar{r}_j^{\nu-1}\right)^2}$.

Each model has been trained on the same training set with 20 samples which gives reservation schedules that are then fixed when testing OOS. Figure 12 show the OOS cost for the model (7) solved normally as a LP and the ADMM model in (12). As seen in the figure, there is little difference between the two approaches and it seems the main difference comes from the reserve cost since the difference in total cost is rather constant across realizations.

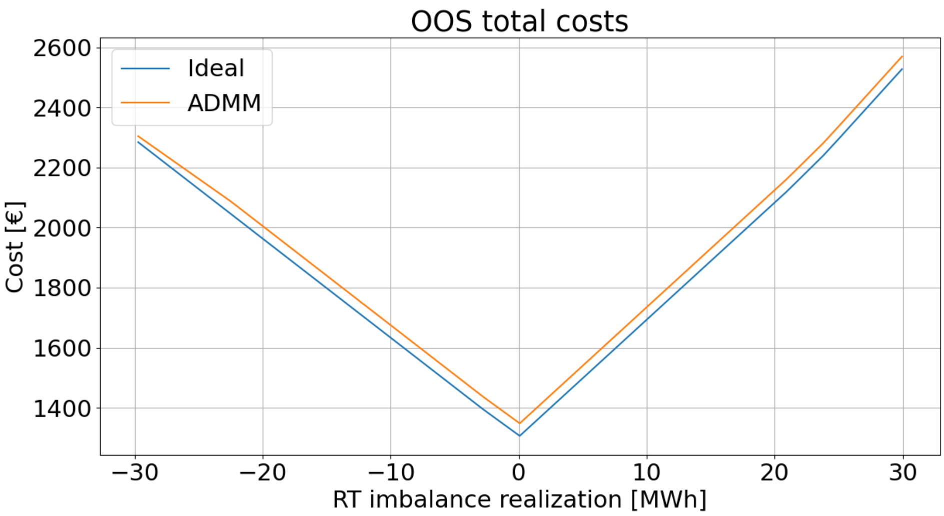


Figure 12

This is confirmed when looking at Table 3 where we can see some differences in the reserve schedules. For example, they disagree significantly on the up reserve allocated to generators 1, 5, 7, and 9. There is also a slightly lower reservation cost for the ideal model, indicating that the ADMM model has not converged fully to the best solution yet. For the stochastic market clearing problem in 7, it takes longer to solve it using ADMM since each scenario in the training phase need to be solved per iteration, i.e. 20 problems per iteration. As mentioned, this can be beneficial for much larger and more complex problems and a distributed framework can be used to solve the subproblems in parallel, such as MIP [5]. Otherwise, asynchronous ADMM is used in practice as well since it removes dependencies on subproblems that are hanging.

Table 3: Generator comparison between ADMM model (12) and ideal stochastic model (7) as well as generator offers.

	ADMM	Ideal	MWh offers	Price offers
G0-up	18	14	18	5
G1-up	9	25	30	7
G2-up	21	21	21	9
G3-up	0	0	3	9
G4-up	6	4	6	5
G5-up	0	15	38	8
G6-up	0	6	13	8
G7-up	45	29	45	3
G8-up	3	3	3	8
G9-up	48	27	48	9
G0-down	0	6	9	5
G1-down	0	1	2	9
G2-down	0	5	18	6
G3-down	17	15	49	5
G4-down	0	11	36	4
G5-down	0	0	28	8
G6-down	0	7	41	7
G7-down	37	22	37	3
G8-down	49	18	49	3
G9-down	0	3	26	4

References

[1] Juan M Morales et al. *Integrating renewables in electricity markets: operational problems.* Volume 205. Springer Science & Business Media, 2013.

[2] *CVXPY.* `https://www.cvxpy.org/index.html`. Accessed: 2021-06-21.

[3] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization.* Cambridge university press, 2004.

[4] Stephen Boyd, Neal Parikh, and Eric Chu. *Distributed optimization and statistical learning via the alternating direction method of multipliers.* Now Publishers Inc, 2011.

[5] *MPI.* `https://mpi4py.readthedocs.io/en/stable/`. Accessed: 2021-08-11.