Scenario generation using historical data paths

Michal Kaut*

January 31, 2020

In this paper, we present a method for generating scenarios by selection from historical data. We start with two models for a univariate single-period case and then extend the better-performing one to the case of selecting sequences of multivariate data. We then test the method on data series for wind- and solar-power generation in Scandinavia.

1 Introduction

There are situations where it is required that the generated scenarios consists of actual data points or sequences from historical data. For example, users of the optimization model may feel more confidence if the model uses 'real' data, instead of synthetic ones. Moreover, using historical sequences ensures correct dependencies both between variables and in time—these would otherwise have to be captured in some way.

In this context, we have to distinguish two situations: in one, we have historical data and want to generate scenarios from the historical distribution, or possibly some subset of it, such as only winter or only week days. This is typical for long-term models that are not used operationally. For example, such series are employed in the TIMES (Loulou and Lettila, 2016; Loulou et al., 2016) and EMPIRE (Skar et al., 2016) energy models.

In operational models, on the other hand, we typically require scenarios that represent the near future, given the current state, i.e., we need to estimate *conditional distributions*.

In this paper, we present an optimization-based method for the former case, i.e., for selecting points or sequences from historical data that represent a good approximation of the empirical distribution.

We start with the simplest case of generating scenarios for single values: we investigate the problem in Section 2, derive several MIP formulations in Section 3 and then test their performance in Section 4. Then, in Section 5, we extend the models for the general case of scenarios consisting of sequences of multivariate parameters. Finally, we test

*SINTEF, Trondheim, Norway; michal.kaut@sintef.no. ORCID 0000-0002-7251-5236.

the extended model using data for wind- and solar-power generation several regions in Scandinavia, in Section 6.

2 Univariate selection problem

In this simplified case, we want to select S values from a univariate data set D containing N data points, such that the empirical distribution of the subset is as close to the empirical distribution of the whole set as possible. This raises two questions: how to measure the closeness of the distributions, and how to find a subset that minimizes the chosen metric.

For the closeness measure, a natural choice seems to be the Kolmogorov-Smirnov statistic, i.e., the supremum of the absolute distance between the two distribution functions. Unfortunately, this distance does not scale well for multivariate data, so it is not suitable for our purpose.

Another choice is the Wasserstein (or Kantorovich–Rubinstein) distance, also known as the "earth mover's distance" in computer science. This metric has a known connection to scenario generation, see for example Pflug (2001); Pflug and Pichler (2011, 2014).

Finally, we can measure the closeness in terms of differences in moments of the marginal distributions and correlations. This approach has also an existing connection to scenario generation, starting with Høyland and Wallace (2001).

Once we have chosen a measure, we have to find a method for identifying a subset that minimizes it. One approach, used in Seljom and Tomasgard (2015), is to randomly select a large number of candidate subsets, evaluate the given measure (they use moments and correlations) on all of them and then chose the one that minimizes it. While this approach works, it provides only statistical guarantees about the quality of the identified subset. Indeed, if there are only few subsets that are significantly better than the rest, they might not be discovered by this approach. Moreover, random selection implies equiprobable scenarios, which limits the achievable match.

For this reason, we propose to use optimization, in particular mixed integer linear programming (MIP), for the task. This is applicable both to the Wasserstein distance (W_p, with $p \in \{1, \infty\}$) and, perhaps surprisingly, the moment based approach—which turns out to be linear because the scenario values are given. For the same reason, it is possible to have the output probabilities as variables in both the MIP models, which should help to achieve a better match. This is an advantage over the 'sample and evaluate' approach from Seljom and Tomasgard (2015).

3 MIP formulation of the univariate selection problem

In the univariate case, we need only two items of input information:

D	vector of data; D_n for $n \in \mathcal{N} = \{1, \ldots, N\}$
S	number of scenarios to generate (data points to select)

The value-selection problem is easily modelled using one set of variables

x_n binary: select D_n or not

and a single constraint

$$\sum_{n \in \mathcal{N}} x_n = S\,. \tag{1}$$

3.1 Minimizing the Wasserstein distance

To minimize the Wasserstein distance, we want to transform the original distribution (with probability mass $1/N$ on every data point) to the scenario distribution (with non-zero probabilities only on S points), by moving as little probability mass as possible. For this, we need the following extra variables (Pflug and Pichler, 2011):

$\pi_{n_1 n_2}$	probability mass moved from D_{n_1} to D_{n_2}
p^{S}_n	probability distribution of the selection

$$\sum_{n_2} \pi_{n_1 n_2} = \frac{1}{N} \qquad n_1 \in \mathcal{N} \tag{2}$$

$$\sum_{n_1} \pi_{n_1 n_2} = p^{\mathrm{S}}_{n_2} \qquad n_2 \in \mathcal{N}\,, \tag{3}$$

where $1/N$ in (2) comes from an assumption that the data points are equiprobable; otherwise, we would simply replace it by the probability of n_1. Now, we have a choice whether we want to fix the output probabilities to $1/S$, or whether we want them free. In the former case, we need to add

$$p^{\mathrm{S}}_n = \frac{1}{S}\, x_n \qquad n \in \mathcal{N}\,, \tag{4}$$

which can be substituted directly into (3). To get free probabilities, we need instead

$$p^{\mathrm{S}}_n \leq x_n \qquad n \in \mathcal{N} \tag{5a}$$

$$p^{\mathrm{S}}_n \geq P^{\min} x_n \qquad n \in \mathcal{N} \tag{5b}$$

$$\sum_{n} p^{\mathrm{S}}_n = 1\,, \tag{5c}$$

where $P^{\min}$ is the minimal allowed probability, required to avoid selected values with zero probabilities. (We can, in addition, add some $P^{\max}$ to (5a), to ensure a more even distribution of probabilities.)

For the objective function, we have a choice between W_1 and W_∞, i.e., between

$$\text{minimize} \sum_{n_1 n_2} \lVert D_{n_1} - D_{n_2} \rVert \cdot \pi_{n_1 n_2} \tag{6}$$

and

$$\begin{aligned} &\text{minimize } d \\ &\quad \text{s.t. } d \geq \lVert D_{n_1} - D_{n_2} \rVert \cdot \pi_{n_1 n_2} \qquad n_1, n_2 \in \mathcal{N} \end{aligned} \tag{7}$$

Since this approach does not guarantee to preserve the mean from the original distribution, and since stochastic-programming models are often sensitive to errors in the mean (Chopra and Ziemba, 1993), we might consider controlling the output sample mean $\bar{d}$, defined as

$$\bar{d} = \sum_n p^{\mathrm{S}}_n \, D_n \, .$$

This can be done either by adding $||\bar{d} - \mu||$ to the objective (with some weight), or by adding the constraint $\bar{d} = \mu$—though the latter should only be used with free probabilities p^{S}, otherwise it is very likely to make the problem infeasible.

3.2 Minimizing the difference in moments

Following Høyland and Wallace (2001), we use the first four moments, that is

mean	$\mu = \mathrm{E}[D]$
variance	$\sigma^2 = \mathrm{E}\big[(D-\mu)^2\big]$
skewness	$\gamma = \mathrm{E}\left[\left(\frac{D-\mu}{\sigma}\right)^3\right] = \mathrm{E}\big[(D-\mu)^3\big] / \sigma^3$
kurtosis	$\gamma = \mathrm{E}\left[\left(\frac{D-\mu}{\sigma}\right)^4\right] = \mathrm{E}\big[(D-\mu)^4\big] / \sigma^4$

The last two formulas are nonlinear due to the scaling by σ, so we use unscaled versions instead. To get the expected values, we model the probabilities the same way as in the previous section, i.e., use (4) for output probabilities fixed to $1/S$, or (5) for free output probabilities. With that in place, we can write the expected value of the m-th power as

$$r_m = \mathrm{E}[D^m] = \sum_n p^{\mathrm{S}}_n \, D^m_n \tag{8}$$

and, using the binomial expansion, the *m-th central moment* as

$$c_m = \mathrm{E}\big[(D-\mu)^m\big] = \sum_{k=0}^{m} \binom{m}{k} (-1)^{m-k} \, \mathrm{E}\big[D^k\big] \mu^{m-k} \, . \tag{9}$$

The first equation is linear in p^{S}_n, since D_n is data. To make the second equation linear in $\mathrm{E}\big[D^k\big]$, and hence also in p^{S}_N, we need μ to be a constant. In other words, we have to replace the actual sample mean by its target value…so the computed moments will be exact only if the sample mean is exactly equal to μ. Alternatively, we can avoid the approximation by matching directly the expected powers $\mathrm{E}\big[D^m\big]$.[1]

The complete MIP formulation using expected powers is then

$$\text{minimize } d \tag{10a}$$

[1] Another way of avoiding the approximation adding $r_1 = \mu$ as a constraint to the model. This, however, could make the model infeasible, especially in the multi-variate case.

subject to:

$$\sum_n x_n = S \tag{10b}$$

$$r_m = \sum_n p_n^{\mathrm{S}}\, D_n^m \qquad m \in \mathcal{M} \tag{10c}$$

$$d_m^+ \geq r_m - P_m \qquad m \in \mathcal{M} \tag{10d}$$

$$d_m^- \geq P_m - r_m \qquad m \in \mathcal{M} \tag{10e}$$

$$d = \sum_m W_m \,(d_m^+ + d_m^-)\; C_m \tag{10f}$$

where $\mathcal{M} = \{1, \ldots, 4\}$ is the set of powers, P_m are the target values of $\mathrm{E}\big[D^m\big]$, C_m are scaling parameters required to make the distances scale-independent, and W_m are weights that allow us to prioritize the moments. The natural choice of C_m is $1/|P_m|$, except when P_m is close to zero. For this reason, we use $C_m = 1/\max(1, |P_m|)$.

Since we usually expect the sensitivity of an optimization problem to decrease with the order of the moment of the input data (see, e.g., Chopra and Ziemba, 1993), it is natural to use a decreasing sequence of weights. In our case, we have used $W = \{10, 5, 2, 1\}$.

Note that the scaling with C_m can be done on lines (10c)–(10e), instead of (10f). This is 'mathematically equivalent', but might exhibit different behaviour numerically. Alternatively, we can avoid the scaling altogether by generating scenarios with $\mu = 0$ and $\sigma = 1$ and transforming the target values to the correct mean and standard deviation in a simple post-processing step.

If we want to match the central moments directly, the MIP formulation becomes

$$\text{minimize } d \tag{11a}$$

subject to:

$$\sum_n x_n = S \tag{11b}$$

$$r_m = \sum_n p_n^{\mathrm{S}}\, D_n^m \qquad m \in \mathcal{M} \tag{11c}$$

$$c_m = \sum_{k=0}^{m} \binom{m}{n} (-1)^{m-k}\, r_k\, \mu^{m-k} \qquad m \in \mathcal{M} \tag{11d}$$

$$d_m^+ \geq c_m - M_m \qquad m \in \mathcal{M} \tag{11e}$$

$$d_m^- \geq M_m - c_m \qquad m \in \mathcal{M} \tag{11f}$$

$$d = \sum_m W_m \,(d_m^+ + d_m^-)/\sigma^m \tag{11g}$$

where M_m are the target values of the central moments, $\mu = M_1$ and $\sigma = \sqrt{M_2}$, and we use $1/\sigma^m$ as the scaling parameter C_m. Note that also in this case we have the option to do the scaling on lines (11d)–(11f), instead of (11g).

4 Numerical tests for the univariate problem

Both the presented MIP models (10) and (11) have N binary variables, one for each input value. If we want this approach to be used in practical applications, we have to be able to solve the models with (at least) several thousands data points in a reasonable amount of time—after all, we are still talking about univariate data.

Since both presented models come in several variants, we want to identify the most promising ones to use in the multivariate case. In addition to the already mentioned variants, we also test the effect of substituting some variables out of the model, or introducing additional ones. For example, we can rewrite (6) as

$$\begin{aligned} &\text{minimize } d \\ &\text{s.t. } d = \sum_{n_1 n_2} \| D_{n_1} - D_{n_2} \| \cdot \pi_{n_1 n_2} \end{aligned}$$

While this is 'mathematically equivalent', it changes the structure of the LP matrix and therefore, potentially, the numerical behaviour of the problem. However, this also makes the test result depend on the solver, as it can be expected that different solvers exploit the matrix structure differently.[2]

All models in the test are implemented in Fico™ Mosel modelling language and solved using Fico™ Xpress version 8.6 on a laptop with Intel® Core™ i7-7600U CPU and 16 GB RAM. All test cases use test data randomly sampled from a normal distribution with $\mu = 1$ and $\sigma = 2$. The tests are repeated 100 times and we use the same data sample for all model variants in each iteration.

4.1 Model using the Wasserstein distance

We consider the following variations of the Wasserstein-distance-based model:

- probabilities p^{S} can be free, or fixed to $1/S$
- whether to use W_1 or W_∞ distance, i.e., whether to minimize the average or the maximum distance
- whether to minimize the sample mean's difference from μ
- whether to fix the sample mean to μ – only if p^{S} are free

Since the third split is only for cases with $p = 1$, this gives in total 10 test variants, summarized in Table 1.

For cases with continuous probabilities, we use $P^{\min} = 1/(\sqrt{10}\,S)$ and $P^{\max} = \sqrt{10}/S$, i.e., we limit the probabilities to

$$\frac{1}{\sqrt{10}\,S}\, x_n \le p_n^S \le \frac{\sqrt{10}}{S}\, x_n \,. \tag{12}$$

[2] This is, of course, true for the test as a whole, but while the other variants represent different models, these ones differ *only* numerically.

Table 1: Tested variants of the model using Wasserstein distance, corresponding to results in Fig. 1.

variant	1	2	3	4	5	6	7	8	9	10
free p^S	✗	✗	✓	✓	✗	✗	✓	✓	✓	✓
use W_∞	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓
opt. mean	✗	✗	✗	✗	✓	✓	✓	✓	✗	✗
fix mean	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓

Table 2: Tested variants of the moment-based model, corresponding to results in Fig. 1.

variant	1	2	3	4	5	6	7	8	9	10
min. moments	✗	✗	✓	✓	✗	✗	✗	✓	✓	✗
early scaling	✗	✓	✗	✓	✗	✗	✓	✗	✓	✗
normalized	✗	✗	✗	✗	✓	✗	✗	✗	✗	✓
free p^S	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓

This implies that the highest probability is at most 10× larger than the smallest one. For cases where we also optimize the mean, we add its difference to the objective with weight equal to one.

4.2 Model using moment matching

For the moment-matching model, we consider the following variations:

- whether to match the expected powers (10) or central moments (11)
- whether to scale the powers/moments in the objective, or in the constraints
- whether to normalize the input, i.e., use $(\mu, \sigma) = (0, 1)$
- whether the probabilities p^{S} are free of fixed to $1/S$.

Note that in the normalized case, there is no difference between expected powers and central moments and no scaling is necessary, which reduces the number of possible combinations somewhat. We end up with 10 variants, presented in Table 2. In the case of free probabilities, we use the limits from (12).

4.3 Results

First, we test both approaches with $N = 100$, and $S \in \{10, 20\}$. We generate 100 different samples and solve each of the variants on them, with time limit set to 300 s. The results are presented in Fig. 1.

For the Wasserstein-distance-based model, we see that variants minimizing the maximum deviation (even case ids) take longer than maximizing the overall deviation. Moreover, cases with fixed probabilities take longer than letting them free—even though

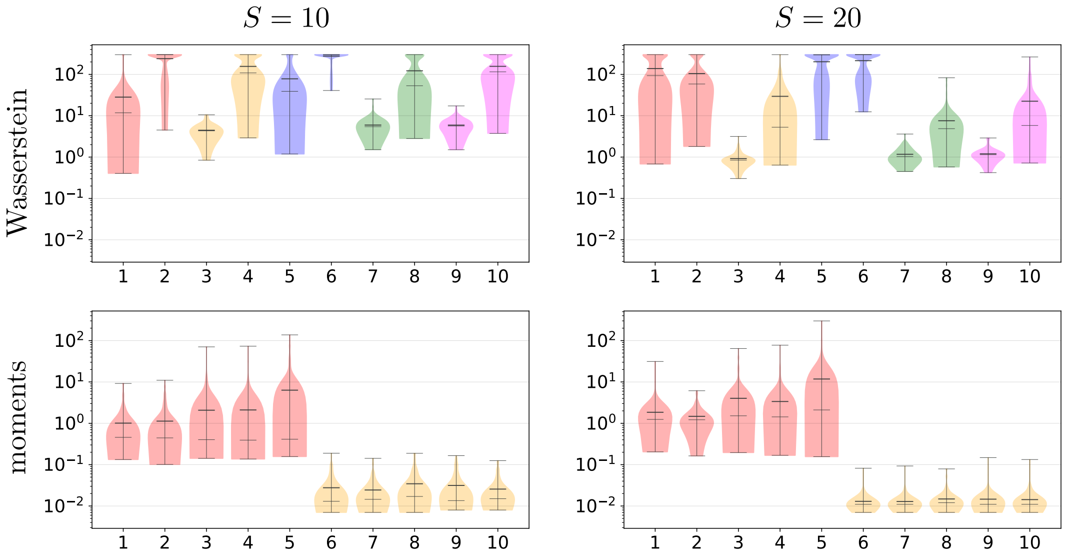

Figure 1: Solution times of model variants defined in Tables 1 and 2, with $N = 100$. For each variant, the *violin plot* estimates the distribution of solution times, based on 100 trials. The horizontal lines in each plot depict the extrema, mean (thicker line) and median.
Note that we a time limit of 300 s, so values at the top of the charts denote cases that did not solve to optimum within that time limit.

the latter needs more variables. This leaves us with three variants, $\{3, 7, 9\}$, that are significantly faster than the rest.

Also for the moment-based model, the variants with free probabilities (6–10) clearly outperform the variants with fixed probabilities—they solve ca. 100 times faster. Apart from the probabilities, there is little difference between the other variants. Based on this, we keep variants 6–10 for further investigation.

Next, we look at the case with $N = 1000$. Here, none of the Wasserstein-distance-based models can no longer be solved within the time limit, probably due to the fact that the number of π_{ij} variables grows quadratically with N. For this reason, we from now on focus solely on the moment-based approach with free probabilities.

We increase the number of tested model variants in order to investigate the effect of substituting some of the decision variables from model (10) and (11). This gives us $4 \times 5 = 20$ new variants, described in Table 3. The results are presented in Fig. 2, for $N \in \{1000, 10\,000\}$. There, we can see that the model takes approximately 0.1 s for $N = 1000$ and 1 s for $N = 10\,000$. In both cases, it is faster to generate 20 scenarios than 10, probably because it is easier to find a good match with more scenarios. With $N = 10\,000$, we can also observe that the variable substitution does indeed make a difference in some model variants, as shown by differences within the coloured groups.

Based on this test, as well as additional tests not reported here, we chose model variant 5 for our subsequent tests. It is, however, important to stress that this choice is almost

Table 3: Tested variants of the moment-based model, used in tests presented in Fig. 2. Variants 17–20 are the same as 1–4, but with normalized distribution.

variant	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
min. moments	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓
early scaling	✗	✗	✗	✗	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓	✓	✓
subst. out r_m	✗	✗	✓	✓	✗	✗	✓	✓	✗	✗	✗	✓	✗	✗	✗	✓
subst. out c_m	–	–	–	–	–	–	–	–	✗	✗	✓	✓	✗	✗	✓	✓
subst. out d	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✓	✗	✓	✓	✓

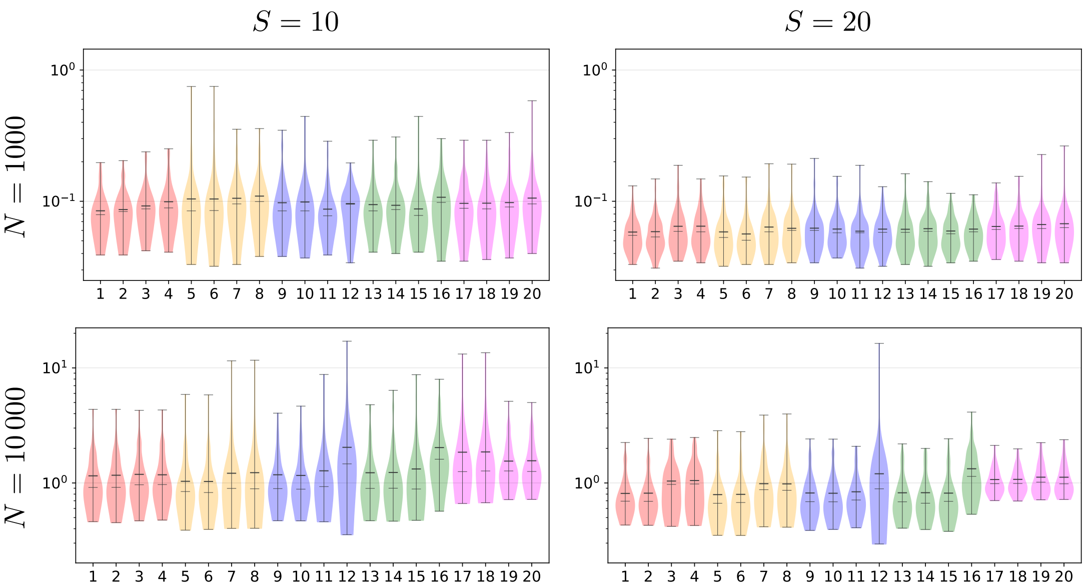

Figure 2: Solution times of the moment-based model with free probabilities, for $N \in \{1000, 10\,000\}$. As in Fig. 1, each violin plot estimates the distribution of solution times for a given model variant (defined in Table 3). In each coloured group, the model variants differ only by some variables being substituted out of the model.
Note the difference in scale between the two rows.

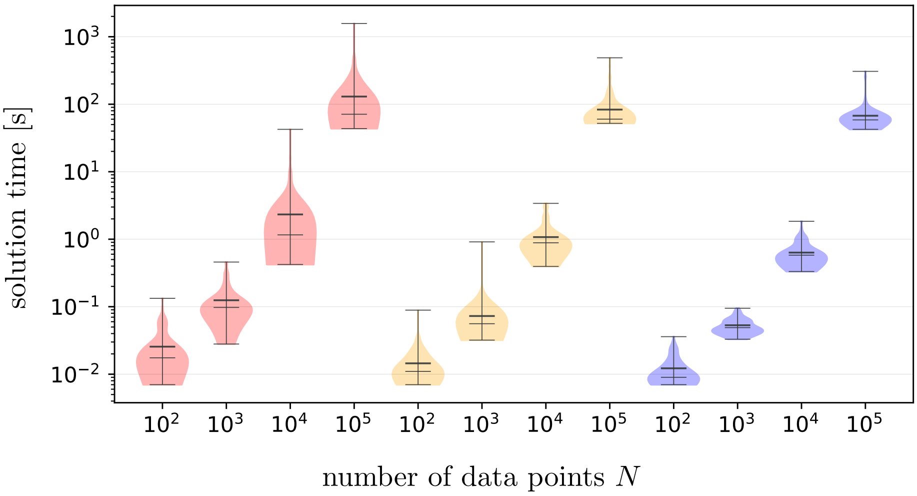

Figure 3: Distribution of solution times for the univariate moment-matching case with free probabilities, using model variant 5 from Table 3 and Fig. 2. The three coloured groups correspond to $S \in \{10, 20, 50\}$.

surely solver-dependent, so it is by no mean a general recommendation.

For this selected variant, Fig. 3 shows how the solution times change with increasing sample size N, for three different values of S. We can see that from $N = 10\,000$ to $N = 100\,000$, the solution time increases almost hundred times, suggesting that this is close to the limit for this method—and this for the simplest, univariate case. As in the previous case, the solution time generally decreases with the number of scenarios S.

The optimal objective values are zero for most cases, and all below 10^{-3}, showing that the free probabilities provide large enough solution space for finding perfect—or at least very good—matches. This, however, is likely to change in the multi-variate case, since there will be many more values to match, without any extra degrees of freedom.

5 Model extensions

In this section, we show how to extend the basic models from the previous section, to handle sequences of multivariate data, instead of single univariate data points. We will describe these extensions only for the moment-based models, because the model using Wasserstein distance turned out to scale poorly. Moreover, both the extensions described in this sections would not change the formulation of the Wasserstein-distance-based model: the only thing that would change is the definition of distances $||D_i - D_j||$.

5.1 Multivariate case

To extend the moment-matching model from (10) or (11), we have to do two things: all elements of the model, except for x_n and d, need to get an additional index $i \in \mathcal{I}$ for the dimension, and all constraints except for the definition of d have to be repeated for all values of the index. For example, (11c) would become

$$p_{mi} = \frac{1}{S} \sum_n x_n \, D^m_{ni} \qquad m \in \mathcal{M}, \, i \in \mathcal{I} \,, \tag{13}$$

where D_{ni} is the i-th component of value D_n. Then, we have to add matching of correlations. The general formula for correlations is

$$\rho_{ij} = \frac{\mathrm{cov}(D_i, D_j)}{\sigma_i \, \sigma_j} = \frac{\mathrm{E}[(D_i - \mu_i)\,(D_j - \mu_j)]}{\sigma_i \, \sigma_j} = \frac{\mathrm{E}[D_i \, D_j] - \mu_i \, \mu_j}{\sigma_i \, \sigma_j} \,,$$

where μ_i and σ_i are, respectively, the sample mean and standard deviation of margin $i \in \mathcal{I}$. Just like for the moments, we have the choice of matching the correlations, with μ_i and σ_i replaced by their target values, or matching only $\mathrm{E}[D_i \, D_j]$.

Starting with the latter, this means updating the model from (10) with

$$r_{ij} = \sum_n p^{\mathrm{S}}_n \, D_{ni} \, D_{nj} \qquad i, j \in \mathcal{I}, i < j \tag{14a}$$

$$d^+_{ij} \geq r_{ij} - P_{ij} \qquad i, j \in \mathcal{I}, i < j \tag{14b}$$

$$d^-_{ij} \geq P_{ij} - r_{ij} \qquad i, j \in \mathcal{I}, i < j \tag{14c}$$

$$d = \sum_m W_m \, (d^+_m + d^-_m) \; C_m + \sum_{i<j} W_{ij} (d^+_{ij} + d^-_{ij}) \,, \tag{14d}$$

where P_{ij} are the target values of $\mathrm{E}[D_i \, D_j]$ and W_{ij} are the optional weights of each individual correlation in the objective.

To match the correlations directly, we need to update (11) with

$$r_{ij} = \sum_n p^{\mathrm{S}}_n \, D_{ni} \, D_{nj} \qquad i, j \in \mathcal{I}, i < j \tag{15a}$$

$$s_{ij} = (r_{ij} - \mu_i \, \mu_j) / (\sigma_i \, \sigma_j) \qquad i, j \in \mathcal{I}, i < j \tag{15b}$$

$$d^+_{ij} \geq s_{ij} - \Sigma_{ij} \qquad i, j \in \mathcal{I}, i < j \tag{15c}$$

$$d^-_{ij} \geq \Sigma_{ij} - s_{ij} \qquad i, j \in \mathcal{I}, i < j \tag{15d}$$

$$d = \sum_m W_m \, (d^+_m + d^-_m) \; C_m + \sum_{i<j} W_{ij} (d^+_{ij} + d^-_{ij}) \,, \tag{15e}$$

where Σ_{ij} are the target correlations.

Just as with the moments, we may consider substituting out some of the variables defined with equality constraints.

5.2 Selecting sequences

Up to now, we have concentrated on selecting a single value from the historical data. The starting point of the paper, on the other hand, was selecting whole sequences from the data, in order to achieve realistic dynamics. In this section, we therefore discuss how to use and/or adapt the presented models to handle time sequences.

Let us say that we have 10 years of hourly data for 5 different variables, and want to select 50 days, such that their distribution is a good approximation of the empirical distributions. In the context of our models, this means $N = 3650$, $\mathcal{I} = \{1, \ldots, 5\}$ and $S = 50$.

One way of solving this with the moment-based model is to do the matching per hour, that is, to add one more index to all the model's variables except x and d. For example, (13) would become

$$r_{mi}^{h} = \sum_{n} p_n^{\mathrm{S}} \, D_{nih}^{m} \qquad m \in \mathcal{M},\, i \in \mathcal{I},\, h \in \{0, \ldots, 23\}\,,$$

where D_{nih} is the value of the i-th component at hour h of day n, and the h superscript of r_{mi}^{h} is an index, no a power. For the correlations, there would probably be no need to control the inter-temporal correlations, since these are guaranteed by selecting whole sequences. In other words, we would simply repeat the matching for each hour. This implies that the number of matched correlations would increase linearly with the length of the sequence (while in the general case we would expect the number of correlations to grow with the square of the dimension).

The above approach does, obviously, increase the size of the model considerably, even if it does not change the number of binary variables. Moreover, it is questionable whether it makes sense to try matching so many parameters, especially with small S. In many cases, there might be some natural aggregate measure that should be enough to match. For example, with inflow series for hydro-models, matching the distribution of total inflows could be enough – while the fact that we use historical data would ensure that the intra-day profiles are realistic. Similarly, if we generate scenarios for wind- or solar-power capacity factor, then it could suffice to match the distribution of the daily average values (so we have a realistic distribution of good and bad days). We could also choose a middle way and do a partial integration, by using only a couple of values per day (day/night, peak/off-peak, etc).

6 Test case

To test the complete method, we generate scenarios for daily sequences for wind- and solar power capacity factors, i.e., production as a fraction of the installed capacity, for use in the TIMES-Norway model (Loulou et al., 2016). We consider several regions in Scandinavia and use a combination of data series from EMHIRES[3] (Gonzalez Aparicio et al., 2016, 2017) and Renewables.ninja[4] (Pfenninger and Staffell, 2016; Staffell and Pfenninger, 2016). This gives us 22 data series, i.e., $\mathcal{I} = \{1, \ldots, 22\}$, all with hourly resolution.

Since there are seasonal differences in patterns of the power production, we separate the data into seasons and generate scenarios for each season separately. This gives us

[3] See `https://setis.ec.europa.eu/EMHIRES-datasets`.

[4] See `https://www.renewables.ninja`.

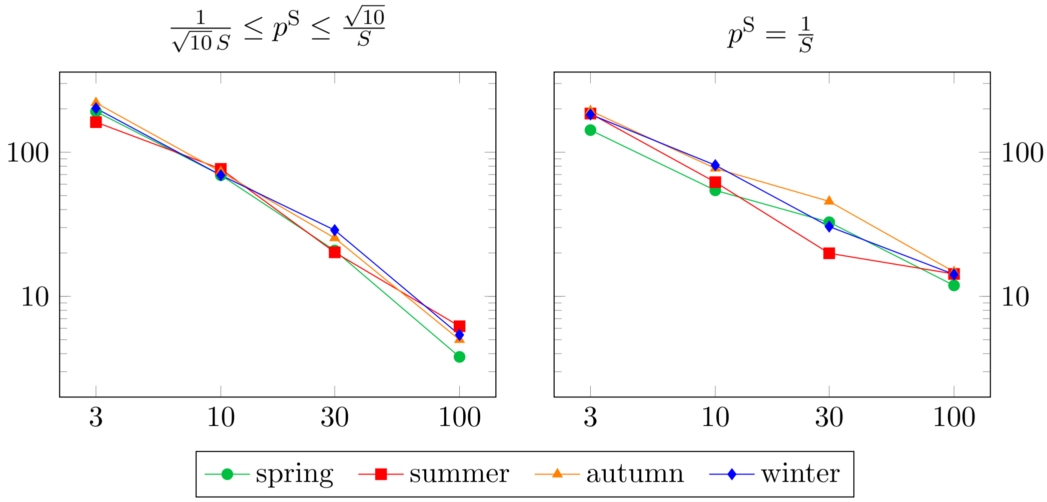

Figure 4: Optimal objective value (total difference) for $S \in \{3, 10, 30, 100\}$, with free and fixed probabilities.

4 data sets, with N varying from 2707 to 2760. We use the moment-matching model to find a subset of days that match the distribution of daily average capacity factors, as discussed in the previous section. The moment weights W_m are $\{10, 5, 2, 1\}$ and all correlations have a weight of 3. In the case of free probabilities, we use the limits from (12), i.e., allow the largest probability to be up to 10 times the smallest one. The model is implemented in Pyomo (Hart et al., 2011, 2017) and solved with the Fico$^{\mathrm{TM}}$ Xpress solver with time limit of 900 s, on the same machine as the tests in Section 4.

We have generated scenarios for $S \in \{3, 10, 30, 100\}$, for each season. Properties of the generated scenarios are summarized in Figs. 4 and 5. The first figure shows that the match improves with increasing S, as expected. It also illustrates the advantage of using free probabilities in order to get a better match, especially for higher S.

Figure 5 provides more details about the matched properties. For example, we can see that means and variances are matched reasonably well already with 3 scenarios and are almost exact with 30 and more scenarios. This is far from obvious result, given that we are talking about matching $2 \times 22 = 44$ values and the only action is selecting probabilities of S vectors out of N possible—and the model is trying to match the other properties at the same time.

The correlations and higher moments are, as expected, more difficult to match: correlations start to stabilize with 30 scenarios and, together with skewness, become reasonable matched with 100 scenarios, while kurtosis might get significant discretization error even then. Again, it is important to realize that we are matching in total $4 \times 22 + 22 \times 21/2 = 319$ properties, so it should be no surprise that it cannot be done with only 100 scenarios—especially since we can only select values, not change them.

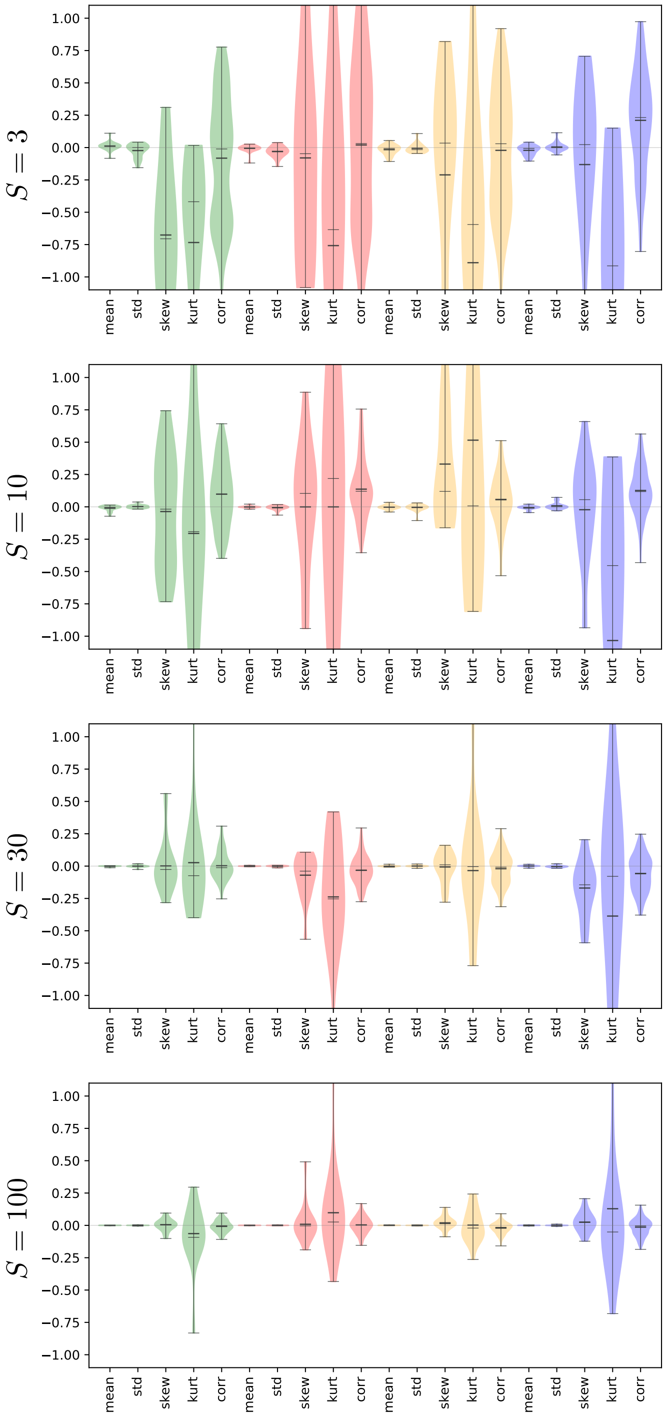

Figure 5: For each case and measured entity, the distribution of differences (mismatch) for that entity, across the modelled variables. The horizontal lines mark the extremes, mean (thick line) and median. The groups denote the seasons, i.e., spring, summer, autumn, and winter.

7 Conclusions

In this paper, we have presented an optimization-based method for selecting representative point or sequences from historical data, in order to obtain a good approximation of the empirical distribution represented by the data. This is needed in cases where the model users do not want to use synthetic values and therefore allow only historical data in scenarios for some optimization model.

We have tested methods based on moment matching and on the Wasserstein distance and found that the moment-based approach scales better with the size of the data, allowing selection from as many as 100,000 data points in the univariate case.

At the moment, the method is limited to generated a set of scenarios, so it is only usable for two-stage optimization models. For multi-stage models, one would need a method that do the selection conditional on current state; this requires a different approach and is left for future research.

Acknowledgements

This work has been done as a part of research project “Assessment of the Value of Flexibility Services from the Norwegian Energy System” (ASSETS), a Norwegian Research Council project no. 268097.[5]

References

V. Chopra and W. Ziemba. The effects of errors in means, variances, and covariances on optimal portfolio choice. *The Journal of Portfolio Management*, 19(2):6–11, 1993.

I. Gonzalez Aparicio, A. Zucker, F. Careri, F. Monforti-Ferrario, T. Huld, and J. Badger. EMHIRES dataset part I: Wind power generation. techreport JRC103442, European Commission, 2016.

I. Gonzalez Aparicio, T. Huld, F. Careri, F. Monforti-Ferrario, and A. Zucker. EMHIRES dataset: Part II: Solar power generation. techreport JRC106897, European Commission, 2017.

W. E. Hart, J.-P. Watson, and D. L. Woodruff. Pyomo: modeling and solving mathematical programs in Python. *Mathematical Programming Computation*, 3(3):219–260, 2011. doi: 10.1007/s12532-011-0026-8.

W. E. Hart, C. D. Laird, J.-P. Watson, D. L. Woodruff, G. A. Hackebeil, B. L. Nicholson, and J. D. Siirola. *Pyomo – Optimization Modeling in Python*, volume 67 of *Springer Optimization and Its Applications*. Springer International Publishing, second edition, 2017. doi: 10.1007/978-3-319-58821-6.

[5] See `https://prosjektbanken.forskningsradet.no/#/project/NFR/268097`.

K. Høyland and S. W. Wallace. Generating scenario trees for multistage decision problems. *Management Science*, 47(2):295–307, February 2001.

R. Loulou and A. Lettila. Stochastic programming and tradeoff analysis in times. Technical report, IEA-ETSAP, 2016. URL `https://iea-etsap.org/index.php/documentation`.

R. Loulou, G. Goldstein, A. Kanudia, A. Lettila, and U. Remme. Documentation for the TIMES model – part I. Technical report, IEA-ETSAP, 2016. URL `https://iea-etsap.org/index.php/documentation`.

S. Pfenninger and I. Staffell. Long-term patterns of european PV output using 30 years of validated hourly reanalysis and satellite data. *Energy*, 114:1251–1265, 2016. doi: 10.1016/j.energy.2016.08.060.

G. C. Pflug. Scenario tree generation for multiperiod financial optimization by optimal discretization. *Mathematical Programming,*, 89(2):251–271, 2001. doi: 10.1007/PL00011398.

G. C. Pflug and A. Pichler. Approximations for probability distributions and stochastic optimization problems. In M. Bertocchi, G. Consigli, and M. A. H. Dempster, editors, *Stochastic Optimization Methods in Finance and Energy*, volume 163 of *International Series in Operations Research & Management Science*, chapter 15, pages 343–387. Springer, 2011. doi: 10.1007/978-1-4419-9586-5_15.

G. C. Pflug and A. Pichler. *Multistage Stochastic Optimization.* Springer Series in Operations Research and Financial Engineering. Springer, 2014. doi: 10.1007/978-3-319-08843-3.

P. Seljom and A. Tomasgard. Short-term uncertainty in long-term energy system models — A case study of wind power in Denmark. *Energy Economics*, 49:157–167, 2015. doi: 10.1016/j.eneco.2015.02.004.

C. Skar, G. Doorman, G. Pérez-Valdés, and A. Tomasgard. A multi-horizon stochastic programming model for the european power system. techreport 2/2016, CenSES, 2016. URL `https://www.ntnu.no/censes/working-papers`.

I. Staffell and S. Pfenninger. Using bias-corrected reanalysis to simulate current and future wind power output. *Energy*, 114:1224–1239, 2016. doi: 10.1016/j.energy.2016.08.068.