Bilevel programming as a means of infinite weighting in regression problems ★

Joakim R. Andersen * **Lars Imsland** *

* *Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway (e-mail: {joakim.r.andersen,lars.imsland}@ntnu.no).*

Abstract: Linear regression is concerned about fitting a model to a set of data. The weighted least squares method is a standard tool for performing linear regression. In this paper, we focus on the case when some of the samples are given priority over others. The residuals for these samples should be given an infinite weighting compared to other samples. However, due to numerical limitations, a weight which is finite but sufficiently large must be chosen instead. We suggest an alternative approach that in practice allows infinite weighting. This is achieved by reformulating the regression optimization problem as a bilevel program. The method is illustrated in a numerical example study. The example shows that, without needing to determine a weighting factor, the proposed method yields the same solution, up to numerical precision, as to the one obtained by using a large weight.

Keywords: Bilevel programming, linear regression, infinite weighting, sample prioritization, daily production optimization,

1. INTRODUCTION

Linear regression is concerned about fitting a model, which is linear in its parameters, to a data set of inputs x_j's and outputs y_j's. The least squares method is a widely used method for linear regression. It finds the parameter vector p that minimizes the squared errors between predictions and measurements

$$\min_p \sum_{j=1}^{n} w_j (y_j - \hat{y}(x_j; p))^2 \tag{1}$$

where $\hat{y}(\cdot)$ is the model and n is the number of samples. In the ordinary least squares method it is assumed that the variance for all samples is constant, yielding $w_j = 1$. An extension of this method is the weighted least squares, where the weights are determined based upon the corresponding estimated variance for the samples, yielding possibly varying w_j's. Seen from a less mathematical perspective, the idea would be to assign higher weights for the samples that one trusts the most. This latter mindset is the one taken in this article.

This work is inspired by a curve estimation task encountered in the oil and gas industry. We look into the case when there are two sources of data regarding modeling of oil flow rates from individual wells. First, the oil flow rate for each well are measured on an infrequent basis. Second, the total oil flow rate for all the wells are measured during production. In this paper, it is assumed that all measurements are taken at steady-states. Further, it is assumed that the output of the total oil rate sensor is of significant higher quality, due to easier measurement principles (single phase flow compared to multiphase flow for the individual wells) and better maintenance as its data is directly related to the amount of sold products and therefore fiscal reporting. Thus, samples from this sensor are considered superior.

In this paper, we propose to use bilevel programming as a tool to achieve prioritization in a weighted linear regression problem for this estimation task.

Bilevel programs are hierarchical optimization problems. There is the *upper-level* problem which has a constraint that involves the solution of another optimization problem. The second optimization problem is referred to as the *lower-level* problem. Both the upper-level and lower-level problems may have their own set of constraints. Bilevel programming is an active field of research, see Dempe (2020) for an overview on theory, algorithms and applications of bilevel optimization.

From the viewpoint of multi-objective optimization, the method we suggest has similarities with the lexicographic method, which also treats different objectives hierarchically.

This paper is organized as follows. In Section 2, the suggested method is presented including approaches to solve the resulting bilevel program. In Section 3, the proposed theory is applied to a small example, and its performance is compared to a weighted optimization problem. Finally, in Section 4, a conclusion is provided.

★ This research is a part of BRU21 – NTNU Research and Innovation Program on Digital and Automation Solutions for the Oil and Gas Industry (www.ntnu.edu/bru21). Financial support from Wintershall Dea Norge AS is acknowledged.

2. BILEVEL PROGRAMMING AS AN INFINITE WEIGHTING SCHEME

In this section, the application of bilevel programming as a means of infinite weighting in regression problems will be presented. The parameter estimation is performed by minimizing the sum of squared residuals in a least squares manner. The formulations are extended to also include constraints. Lastly, strategies to solve the resulting bilevel program are provided.

A common parameter estimation situation would be the need to fit a model to a set of data. In this paper, we are looking into a situation where there are two models, two data sets, and three sets of parameters. Each model may have its own parameters, and in addition, the two models share some, or all, parameters. The two models are:

$$\hat{y}_p(\hat{x}) := \hat{y}(\hat{x}; p_s, \hat{p}) \tag{2a}$$
$$\tilde{y}_p(\tilde{x}) := \tilde{y}(\tilde{x}; p_s, \tilde{p}) \tag{2b}$$

where $p_s \in \mathbb{R}^{n_{p_s}}, \hat{p} \in \mathbb{R}^{n_{\hat{p}}}$ and $\tilde{p} \in \mathbb{R}^{n_{\tilde{p}}}$ are tunable parameter vectors, p_s are shared between the two models, and the $\hat{x} \in \mathbb{R}^{n_{\hat{x}}}$ and $\tilde{x} \in \mathbb{R}^{n_{\tilde{x}}}$ are input vectors. To keep the notation clean, the shorter and slightly misleading left hand side of (2) is used. Further, it is assumed that both models map to scalar outputs:

$$\hat{y}_p : \mathbb{R}^{n_{\hat{x}}} \times (\mathbb{R}^{n_{p_s}} \times \mathbb{R}^{n_{\hat{p}}}) \to \mathbb{R} \tag{3a}$$
$$\tilde{y}_p : \mathbb{R}^{n_{\tilde{x}}} \times (\mathbb{R}^{n_{p_s}} \times \mathbb{R}^{n_{\tilde{p}}}) \to \mathbb{R}. \tag{3b}$$

If there were no shared parameters, *i.e.*, $n_{p_s} = 0$, then these two models could be treated separately, *e.g.*, by solving two least squares problems. A straightforward extension of that idea to the case with $n_{p_s} \neq 0$ would be to solve a weighted least square problem:

$$\min_{p_s, \hat{p}, \tilde{p}} \hat{w} \sum_{j=1}^{m_{\hat{y}}} (\hat{y}_j - \hat{y}_p(\hat{x}_j))^2 + \tilde{w} \sum_{j=1}^{m_{\tilde{y}}} (\tilde{y}_j - \tilde{y}_p(\tilde{x}_j))^2 \tag{4}$$

where the subindex j indicates it is a data point. The values of the weights $\hat{w}$ and $\tilde{w}$ will have a significant impact on the final parameters. The focus of this paper will be where one of these two will have priority over the other. This corresponds to setting the corresponding weight "infinitely large". However, due to limitations in numerical computing, it may be challenging to find a value which gives a reliable desired behavior.

Another way of phrasing the desire of prioritizing one over the other, is to say that the second curve fitting may only happen in the subspace of those directions which does not impact the first curve fitting. Such a subspace may for example exist if there are few data points for the one with highest priority. In this case, the remaining degrees of freedom of the shared parameters(?) may be taken by the other minimization objective. This idea can be posed as a bilevel program:

$$\min_{p_s, \hat{p}} \sum_{j=1}^{m_{\hat{y}}} (\hat{y}_j - \hat{y}_p(\hat{x}_j))^2 \tag{5a}$$
$$\text{s.t.} \quad \min_{p_s, \tilde{p}} \sum_{j=1}^{m_{\tilde{y}}} (\tilde{y}_j - \tilde{y}_p(\tilde{x}_j))^2 \tag{5b}$$

where the lower-level problem (5b) has the highest priority.

To make the parameter estimation procedure more flexible, constraints may be added to the upper and lower problem:

$$\min_{p_s, \hat{p}} \sum_{j=1}^{m_{\hat{y}}} (\hat{y}_j - \hat{y}_p(\hat{x}_j))^2 \tag{6a}$$
$$\text{s.t.} \quad \hat{h}(\hat{x}; p_s, \hat{p}) \geq 0 \tag{6b}$$
$$\hat{g}(\hat{x}; p_s, \hat{p}) = 0 \tag{6c}$$
$$\min_{p_s, \tilde{p}} \sum_{j=1}^{m_{\tilde{y}}} (\tilde{y}_j - \tilde{y}_p(\tilde{x}_j))^2 \tag{6d}$$
$$\text{s.t.} \quad \tilde{h}(\tilde{x}; p_s, \tilde{p}) \geq 0 \tag{6e}$$
$$\tilde{g}(\tilde{x}; p_s, \tilde{p}) = 0 \tag{6f}$$

where (6b) and (6e) also allows for setting bounds on the variables. The interpretation that the upper-level problem is minimized in the subspace defined by the remaining degrees of freedom of the lower-level problem is not completely valid any longer as the constraints (6b)-(6c), which contain p_s, may restrict the lower-level's feasible area. However, the lower-level's objective function still has a higher priority than the upper-level one.

2.1 Solving the bilevel program

A common approach to tackle this type of problem is to replace the lower-level optimization problem by its KKT conditions, as suggested by Fortuny-Amat and McCarl (1981). In the rest of this paper, it is assumed that the lower-level problem is convex and that the Linear Independence Constraint Qualification (LICQ) (Nocedal and Wright, 2006) holds at the optimum. In this case, the KKT conditions are both necessary and sufficient conditions of optimality. The objective functions will be convex for any models that are linear in the parameters. It should be pointed out that the linearity is only required for the parameters and not the variables. For example, a polynomial model of any degree will satisfy this criteria. The KKT conditions for the lower-level problem:

$$\nabla_{p_s, \tilde{p}} \mathcal{L}(\tilde{x}, p_s, \tilde{p}, \lambda_{\tilde{h}}, \lambda_{\tilde{g}}) = 0 \tag{7a}$$
$$\tilde{h}(\tilde{x}; p_s, \tilde{p}) \geq 0 \tag{7b}$$
$$\tilde{g}(\tilde{x}; p_s, \tilde{p}) = 0 \tag{7c}$$
$$\lambda_{\tilde{h}} \geq 0 \tag{7d}$$
$$\lambda_{\tilde{h}} \odot \tilde{h}(\tilde{x}; p_s, \tilde{p}) = 0 \tag{7e}$$

where $\odot$ represents element-wise multiplication, λ_{hl} and λ_{gl} are the Lagrange multiplier vectors, and the Lagrangian is defined as:

$$\begin{aligned} \mathcal{L}(\tilde{x}, p_s, \tilde{p}, \lambda_{\tilde{h}}, \lambda_{\tilde{g}}) = & \sum_{j=1}^{m_{\tilde{y}}} (\tilde{y}_j - \tilde{y}_p(\tilde{x}_j))^2 \ldots \\ & - \lambda_{\tilde{h}}^T \tilde{h}(\tilde{x}; p_s, \tilde{p}) \ldots \\ & - \lambda_{\tilde{g}}^T \tilde{g}(\tilde{x}; p_s, \tilde{p}). \end{aligned} \tag{8}$$

The final optimization problem becomes:

$$\min_{p_s,\hat{p},\tilde{p},\lambda_{\tilde{h}}} \sum_{j=1}^{m_{\hat{y}}} (\hat{y}_j - \hat{y}_p(\hat{x}_j))^2 \tag{9a}$$

$$\begin{aligned}
\text{s.t.}\quad & \hat{h}(\hat{x}; p_s, \hat{p}) \geq 0 && \text{(9b)}\\
& \hat{g}(\hat{x}; p_s, \hat{p}) = 0 && \text{(9c)}\\
& \nabla_{p_s,\tilde{p}} \mathcal{L}(\tilde{x}, p_s, \tilde{p}, \lambda_{\tilde{h}}, \lambda_{\tilde{g}}) = 0 && \text{(9d)}\\
& \tilde{h}(\tilde{x}; p_s, \tilde{p}) \geq 0 && \text{(9e)}\\
& \tilde{g}(\tilde{x}; p_s, \tilde{p}) = 0 && \text{(9f)}\\
& \lambda_{\tilde{h}} \geq 0 && \text{(9g)}\\
& \lambda_{\tilde{h}} \odot \tilde{h}(\tilde{x}; p_s, \tilde{p}) = 0. && \text{(9h)}
\end{aligned}$$

The optimization problem in (9) is a Nonlinear Program (NLP), and a regular NLP solver may be used. However, the constraint (9h), which arises from the complementary conditions of the first order conditions of optimality of the lower-level problem, may be a tough challenge for the solver.

2.2 The complementary condition

The complementary condition, *e.g.*, $\lambda_i \cdot h_i = 0$, says that either λ_i or h_i is zero, or both. However, both may not be nonzero simultaneously. In the bilevel literature, there exists several ways to handle (9h). The two approaches given next were suggested by Fortuny-Amat and McCarl (1981).

Big-M reformulation The complementary condition can be formulated using a binary variable. If the integer variable takes the value $z_i = 1$, then the constraint is active, and a value of 0 indicates it is inactive.

$$\lambda_i \cdot h_i = 0 \iff \begin{cases} h_i \leq M_1(1 - z_i) \\ \lambda_i \geq 0 \\ \lambda \leq M_2 z_i \end{cases} \tag{10}$$

where $z_i \in \{0, 1\}$, and M_1 and M_2 are chosen "large enough" such that desirable behavior is achieved. If there is a known finite upper value of h_i, then this may be used as M_1. The other constant, M_2, is more challenging to determine.

Special Ordered Set of Type 1 Another approach, which avoids the determination of any big-M, involves categorizing variables within Special Ordered Sets of Type 1 (SOS1). At most one variable within a SOS1 may be nonzero. For each complementary condition, a SOS1 is created containing two variables: $\{h_i, \lambda_i\}$, and the solver must be informed of these sets.

3. EXAMPLE STUDY

The methodology was inspired by a part of the daily production optimization (DPO) challenge within the oil and gas industry. The DPO focuses on utilizing the production system in an optimal manner. The goal is to increase revenue while obeying constraints arising from the facility. See Fig. 1 for an illustrative setup. The fluid flow from a well is typically a mixture of oil, gas and water which must be separated into three single-phase streams by the facility. As a simple example of a DPO, the revenues come from selling the oil and gas, whereas the constraints could be on how much water and gas the facility may process.

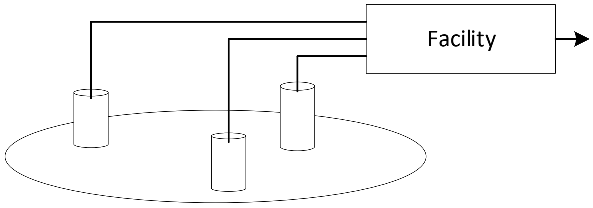

Fig. 1. An example of a production network with one reservoir, three wells, and a production facility.

There are (at least) two different categories of oil flow rates. These will be discussed next.

First, a well's fluid production may be routed to what is known as a test separator. From the data collected at the test separator, the relationship between the downhole pressure, *i.e.* the pressure at the wellbore, and the oil flow rate can be estimated for that well. Because of a limitation on the availability of the test separator, the most recent test data for a well may be old.

Second, the total oil production, *i.e.* the oil production of all the wells, is measured during operation This measurement is often more accurate as it is both newer/more frequent and it is used to determine how much oil the platform exports which needs to be reported for fiscal reasons. The measurements of the downhole pressures for the wells are also available during operation.

Both for the well-based and total measurements, it is assumed that the wells must be in a steady-state before measurements can be taken. This may imply that there will be few recordings of the total oil production.

The problem that motivated this paper was the challenge of merging these two sources of measurements. In the rest of this section, we will study a constructed example where the proposed method is applied. The study focuses on utilizing the available data to estimate the relationship between the downhole pressure and gaslift rate for the wells.

3.1 Setup

In this setup, we have chosen to look at three wells. Each well has a linear model of its relationship between downhole pressure and oil production:

$$\hat{q}_i(P^w) = \alpha_i(P^w_{i,0} - P^w) + \beta_i \tag{11}$$

where α_i and β_i are the two to-be-determined parameters, and $\hat{q}_i(P^w)$ is the predicted oil production for well i at the downhole pressure P^w. Finally, β_i is the predicted oil production at downhole pressure $P^w_{i,0}$. $P^w_{i,0}$ is set to 120,100 and 110 bar for the three wells. For each well test, it is assumed three data points. These points are given in Fig. 2 as the points marked by blue circles.

The upper-level objective function for the bilevel formulation in (6) is set as the sum of the squared prediction errors for the well tests of all three wells:

$$J_{ul} = \sum_{i=1}^{3}\sum_{j=i}^{3}\left(\hat{q}_i(P_{i,j}^{w}) - q_{i,j}\right)^2 \tag{12}$$

where i and j indicates the well and sample number, respectively.

The prediction of the total oil rate is a sum of these individual models:

$$\hat{q}_{\text{tot}}(\mathbf{P}_j) = \sum_{i=1}^{3}\hat{q}_i(P_{j,i}). \tag{13}$$

with $\mathbf{P}_j = [P_{j,1}, P_{j,2}, P_{j,3}]$. The lower-level objective function in (6) is set as the sum of the squared prediction errors of the total oil production:

$$J_{ll} = \sum_{j=1}^{n_{sl}}\left(\hat{q}_{\text{tot}}(\mathbf{P}_j) - q_{\text{tot},j}\right)^2 \tag{14}$$

where n_{sl} is the number of samples of the total oil rate. These samples are hereafter referred to as *system-level* samples. The system-level samples used in this example were arbitrarily chosen and are given in Tab. 1. These samples are assumed to be perfect. *I.e.*, both the sensors for downhole pressures and the sensor for the total oil rate have no type of measurement error.

Table 1. The table shows the different operation points for the system-level samples. Each column gives the data for one operation point, where the second to fourth rows give the downhole pressures in bar, and the last row tells the total oil production in Sm3/d.

j	1	2	3	4	5
Well 1 ($P_{j,1}$)	117	96	100	100	90
Well 2 ($P_{j,2}$)	118	96	104	100	90
Well 3 ($P_{j,3}$)	119.5	99	108	100	95
Oil ($q_{\text{tot},j}$)	17.08	62.50	48.33	56.33	73.83

In this setup, there are only shared parameters, $n_{\tilde{p}} = n_{\hat{p}} = 0$ and $n_{p_s} = 6$. In the first example, there are no constraints.

The optimization formulations were formulated by using CasADi (Andersson et al., 2019) 3.5.5. The nonlinear programs were solved by Ipopt (Wächter and Biegler, 2006). The mixed integer nonlinear programs in Section 3.4 were solved by BONMIN (Bonami et al., 2008). No changes were made to the default settings of the solver.

3.2 Results

In the example, we started without any system-level samples in the curve fitting, and then added one sample at a time. First, the first column of Tab. 1 was added, then the second, and so on. The results are given in Fig. 2. For each well, we have plotted, in dashed-style, the resulting estimated linear relationships provided an increasing availability of system-level samples. The small numbers with arrows indicates the amount of system-level samples that was used in the estimation of that line. The graphs resulting from using 3,4 or 5 system-level samples all lie on top of each other. According to the figure, the slope of all the final lines are the same as those of the red lines. Comparing the blue lines, which are estimated purely based on the well-test data, with the dashed lines, it can

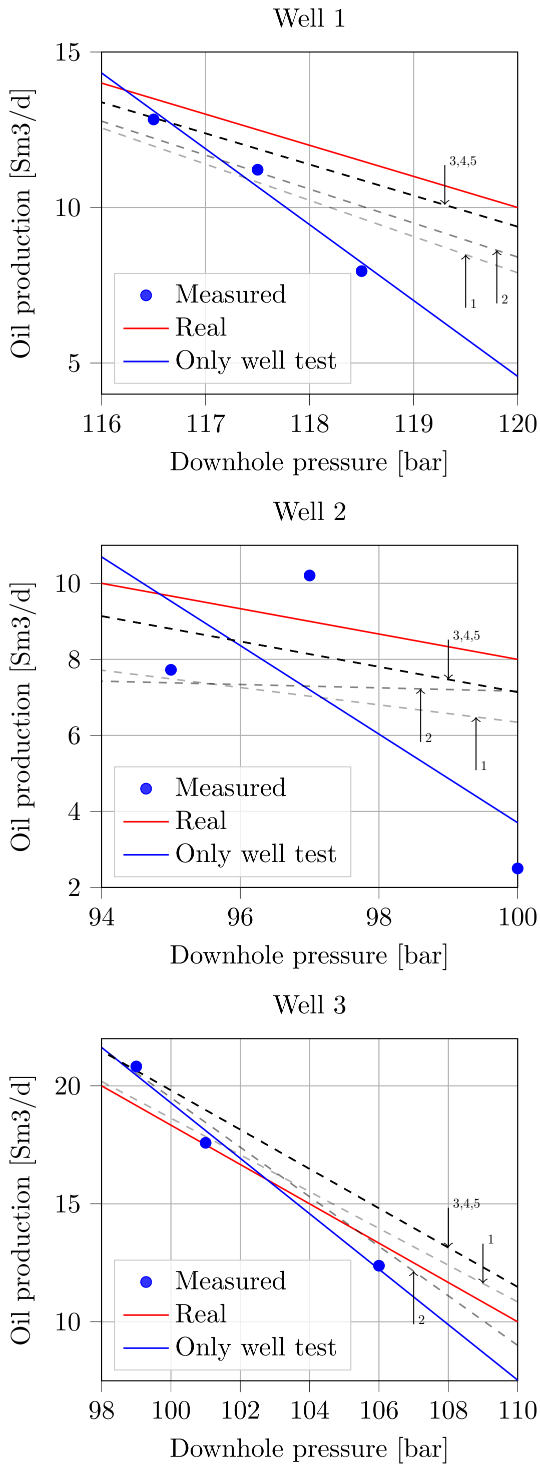

Fig. 2. The blue filled circles are points measured during well tests. The blue lines are the linear relationship given only well-test data. The red lines are the real (unknown) relationships. The dashed lines are the estimated linear relationships given more and more system level points. The small numbers indicates the number of system level points used in the estimation.

be concluded that all the dashed lines provide a better approximation of the slopes of the red lines.

It should be pointed out that the biases, the β_i's, are not perfect. This is to be expected as the system-level samples does not provide information on the value of the individual β_i's, only the sum of them. Having system-level samples where some of the wells are shut and other open, should improve the bias estimation. However, for the current example with 3 (or more) system-level samples the result is that the α_i's are determined by the lower-level problem, and also the sum of the biases. Then the distribution of this sum to the individual β_i's happens through the minimization of the upper-level objective function which focuses on the well-test data.

In Fig. 3, the prediction of the total oil rate is plotted at a test point. The test point is $[119.5, 99.5, 109.5]$ bar which was chosen arbitrary except that it does not coincide with any of the system-level samples. The fact that the prediction does not change by adding more than 3 points can be explained by the lack of measurement error or noise in the total oil rate and the downhole pressures. From Tab. 1, it can be seen that any system-level point after the three first points must be a linear combination of the three previous ones. Thus, no more information can be extracted from the system for this setup unless wells are allowed to be shut.

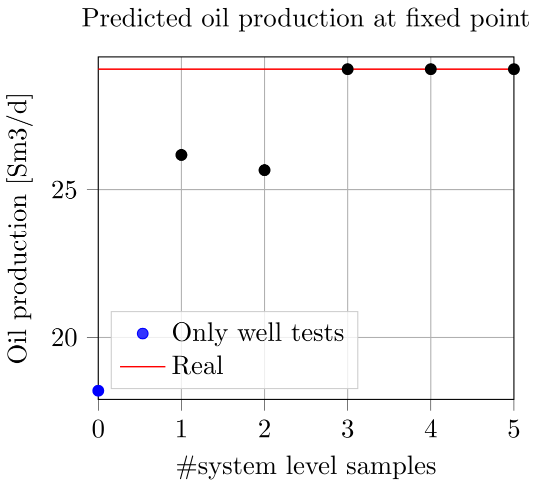

Fig. 3. The red line is the real oil production at operation point $[119.5, 99.5, 109.5]$ bar. The blue dot is the prediction done with only well-test data, whereas the black also contains system level points. The number of samples is given on the x-axis.

3.3 Comparison with weighted optimization

As mentioned in Section 2, the suggested approach can be viewed as a way of performing infinite weighting in a weighted objective function. *E.g.*, $\hat{w} = 1$ and $\tilde{w} = \infty$ in (4). Here we will illustrate by example that solving this weighted minimization problem with an increasing $\tilde{w}$ will yield an almost identical solution as the one obtained by the bilevel programming. In this test, all the well-test samples and the system-level samples are used.

In Fig. 4 the value of (14) is plotted for different values of $\tilde{w}$. The weight start at $\tilde{w} = 10$ and increases by a factor of 10 up to the final value $\tilde{w} = 10^6$. The correct value of J_{ll}, the y-axis value, is zero. It should be zero because there is no measurement noise on the system-level samples, and both the model and the real relationship are linear. In other words, the prediction of the total oil rate should be perfect at the operation points given in Tab. 1. From the figure, it can be seen that the J_{ll} keeps decreasing towards zero as the weight increases. Notice that the scales are logarithmic. The J_{ll} from the weighting through bilevel approach gave a value of "numerically" 0, $\mathcal{O}(10^{-25})$.

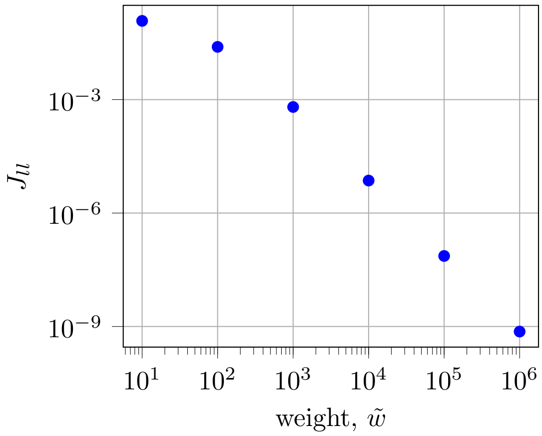

Fig. 4. The objective function value of the lower-level problem converges towards the one resulting from the bilevel approach as the weighting factor increases.

As a side note, When increasing the weight to 10^{14}, the solver fails. This is reasonable as the problem is ill-conditioned due to scaling. However, it highlights another weakness one may encounter when trying to find a "sufficiently" large weight.

To illustrate that the parameters found through the weighted optimization, with an increasing weight, converges towards those found by the bilevel program, the norm between the parameters are shown in Tab. 2. The norm decreases as the weight increases.

Table 2. The table shows the ℓ_2-norm of the difference between the parameters obtained from solving the bilevel program and a weighted problem with the weight given in the first column. In the third column, the value of the lower-level objective function (14) is given.

Weight	ℓ_2-norm	J_{ll}
$1.0 \cdot 10^{+01}$	$1.85 \cdot 10^{+00}$	$1.21 \cdot 10^{-01}$
$1.0 \cdot 10^{+02}$	$9.26 \cdot 10^{-01}$	$2.48 \cdot 10^{-02}$
$1.0 \cdot 10^{+03}$	$1.5 \cdot 10^{-01}$	$6.39 \cdot 10^{-04}$
$1.0 \cdot 10^{+04}$	$1.6 \cdot 10^{-02}$	$7.24 \cdot 10^{-06}$
$1.0 \cdot 10^{+05}$	$1.61 \cdot 10^{-03}$	$7.33 \cdot 10^{-08}$
$1.0 \cdot 10^{+06}$	$1.61 \cdot 10^{-04}$	$7.34 \cdot 10^{-10}$

3.4 Example with constrained bias parameter

In this example the proposed method of weighting through bilevel programming is applied on a similar setup as in Section 3.1. Everything remains the same except that there is introduced a constraint on the bias parameter belonging to Well 2: $\beta_2 \geq 7.8$. The real bias value is 8. All the well-test samples and system-level samples are used for the curve fitting.

The Big-M method mentioned in Section 2 was used to deal with the complementary condition. The M's in (10) were set to $M_1 = M_2 = 100$. This value was large enough to give the correct behavior.

The resulting linear predictors are shown in Fig. 5. The bias parameter of Well 2 is at the given constraint: $\beta_2 = 7.8$. As before, the slopes of the estimated lines and the red lines are all the same according to the figure. The lines for both Well 1 and Well 3 is shifted downwards. By looking at the figures, one may be convinced that these two shifts do indeed reduce the prediction error with respect to the well-test data; the green lines are moved "closer to" the blue circles.

4. CONCLUSION

This paper presented an application of bilevel programming where the formulation was used to introduce infinite weighting. In the studied example, we saw that the proposed method allows for prioritization without having to determine a weighting factor. A disadvantage of the proposed method, compared to the weighting approach, is that integer decision variables will be introduced if there are constraints on the lower-level problem and a Mixed Integer Nonlinear Program (MINLP) solver is required instead of an NLP solver. MINLP solvers are typically slower than its integer-free counterpart. Nonetheless, the method removes issues with ill-conditioned formulations due to a large weight. Further, as no weight is used, the tuning process is eliminated.

REFERENCES

Andersson, J.A.E., Gillis, J., Horn, G., Rawlings, J.B., and Diehl, M. (2019). CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1), 1–36. doi: 10.1007/s12532-018-0139-4.

Bonami, P., Biegler, L.T., Conn, A.R., Cornuéjols, G., Grossmann, I.E., Laird, C.D., Lee, J., Lodi, A., Margot, F., Sawaya, N., and Wächter, A. (2008). An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization*, 5(2), 186–204. doi:https://doi.org/10.1016/j.disopt.2006.10.011. URL `https://www.sciencedirect.com/science/article/pii/S1572528607000448`. In Memory of George B. Dantzig.

Dempe, S. (2020). Bilevel optimization: theory, algorithms, applications and a bibliography. In *Bilevel Optimization*, 581–672. Springer.

Fortuny-Amat, J. and McCarl, B. (1981). A representation and economic interpretation of a two-level programming problem. *The Journal of the Operational Research Society*, 32(9), 783–792. URL `http://www.jstor.org/stable/2581394`.

Nocedal, J. and Wright, S. (2006). *Numerical optimization*. Springer Science & Business Media.

Wächter, A. and Biegler, L.T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1), 25–57.

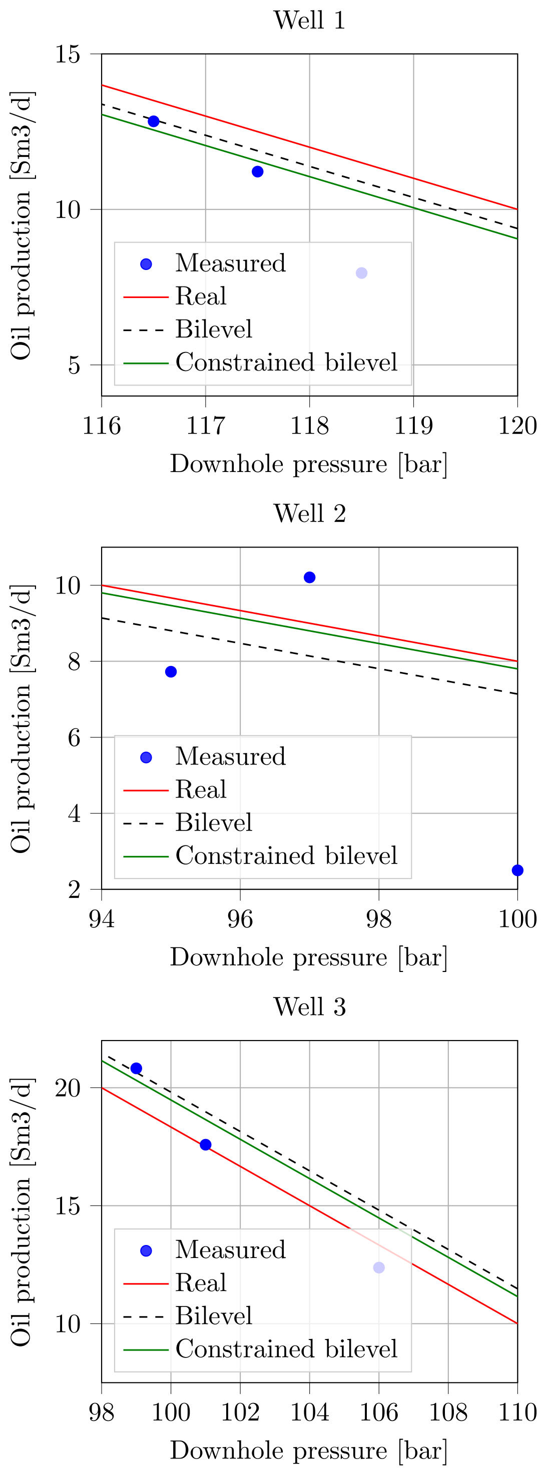

Fig. 5. The blue filled circles are points measured during well tests. The red lines are the real (unknown) relationships. The dashed lines are those obtained with no constraints. The green lines were estimated with the constraint $\beta_2 \geq 7.8$
.