Automated Calibration of Farm-Sale Mixed Linear Programming Models using Bi-Level Programming

Wolfgang Britz
University of Bonn

Abstract

We calibrate Linear and Mixed Integer Programs with a bi-level estimator, minimizing under First-order-conditions (FOC) conditions a penalty function considering the calibration fit and deviations from given parameters. To deal with non-convexity, a heuristic generates restart points from current best-fit parameters and their means. Monte-Carlo analysis assesses the approach by drawing parameters for a model optimizing acreages under maximal crop shares, a land balance and annual plus intra-annual labour constraints; a variant comprises integer based investments. Resulting optimal solutions perturbed by white noise provide calibration targets. The approach recovers the true parameters and thus allows for systematic and automated calibration.

Keywords

linear programming; mixed linear programming; calibration; bi-level programming; farm-scale model

1 Introduction

A stronger focus of agricultural policy on agri-environmental interactions and on the differentiated farm-scale impacts of policy instruments has renewed the interest in programming models (cf. BRITZ et al., 2012). Here, two strands can be observed in literature. The first builds on Positive-Mathematical Programming (PMP, HOWITT, 1995, for a review see HECKELEI et al., 2012). By introducing non-linearities, PMP allows for interior solutions and calibration of models where the number of binding constraints is lower than the number of non-zero variables subject to calibration. That led to PMP based programming models with a quite limited constraint set. Such models operate at regional level (cf. BRITZ and WITZKE, 2012), at the level of farm types inside regions (cf. GOCHT and BRITZ, 2011) or at single farm level (cf. LOUHICHI et al., 2015), covering small regions or even whole Europe (BRITZ and WITZKE, 2012; LOUHICHI et al., 2015). The simulation behaviour of PMP based models is to a large degree determined by the non-linear terms. This motivates a body of literature dealing with estimating and calibrating PMP type models (cf. MEREL and BUCARAM, 2010; HOWITT et al., 2012), including efforts to bridge the gap between econometric work and programming models (cf. JANSSON and HECKELEI, 2011; ARATA and BRITZ, 2019). The estimation of parameters of programming models in their FOC is termed "Econometric Mathematical Programming" (EMP) by BUYSSE et al. (2007), for a review see DE FRAHAN (2019), the cited applications estimate dual cost functions. This paper links to this body of literature by exploring the application of EMP to Linear Programming (LP) and Mixed Linear Programming (MIP) models.

Whereas PMP requires non-linearities, LP and MIP approaches are more common for detailed so-called bio-economic farm-scale models, for a review see JANSSEN and ITTERSUM (2007). Here, the allocative response depends solely on the interplay between the linear objective and the linear constraints: as any (not degenerate) solution is always on the corner of the constraint set, changes in a simulation imply a jump from one such corner to another. Furthermore, in any LP solution, the number of variables away from their bounds is at most as high as the number of binding constraints. A LP hence requires a rich constraint set to avoid highly specialized solutions. Integer variables, for instance to depict indivisibilities in investments, are common in this type of model.

The seminal paper by HOWITT (1995) on PMP focuses on the FOC during calibration of programming models, i.e. calibration reduces or even completely removes differences between marginal costs and revenues. Here, a two-step approach is common both in price endogenous market and farm-scale models. First, data are corrected to fit into the model's constraints. While data balancing for market-scale models requires typically advanced balancing techniques (cf. RODRIGUEZ, 2014; BRITZ, 2021), farm scale models rely mostly on simpler approaches such as scaling resource coefficients to remove infeasibilities. A second step, often called benchmarking, choses parameters such that production and behavioural functions replicate the observed data. Here, certain parameters are considered as fix and given, such as substitution elasticities. Others

DOI: 10.30430/70.2021.3.165-181

are then chosen such that behavioural and productions functions replicate observed quantities at observed prices, in case of PMP, entries in the objective function.

In opposite to the growing literature body on PMP, little has been published on calibration of constraint-rich farm-scale models, which might also reflect their declined use after the emergence of PMP. TROOST and BERGER (2014) report in detail on a systematic approach not applying PMP. JAYET et al. (2020) in their model documentation describe the calibration of the Europe-wide farm type LP Aropaj as "The calibration algorithm is based on sequential calculations, combining Monte-Carlo and gradient methods. In practice, randomization of calibration parameters alternating with 'local' gradient based improvement of the criterion is generated from a large number of LP runs for each of AROPAj farm groups." While the authors report which parameters are subject to calibration (maximal cropping shares, feed requirement, livestock life cycle) and how squared differences from observed acreages, herd sizes and feed use are weighted to define the penalty to minimize, the reference above is all detail given on the algorithm itself, beside reporting that between 1,000 and 2,000 runs of each LP are performed.

We develop in here a generic bi-level based programming approach which penalizes deviations from a-priori distributions both of parameters and of error terms between simulated and observed values. The paper is organized as follows. We next present an algorithmic approach discussing how to systematically remove infeasibilities, to calibrate the model and to avoid degenerate solutions resulting from calibration. The approach estimates (selected) parameters of a LP or MIP under some maximizing behaviour. We follow with applications to two smaller farm-scale didactic models, a LP and MIP. We test and assess the algorithm on variants of these models, solved for one or simultaneously for multiple years. In the latter case, degrees of freedoms do not longer allow for perfect calibration. Specifically, we randomly generate parameters for the model, add normal distributed white noise of different variance to its optimal solution, and finally use the algorithm to calibrate the model against these observations. We assess the achieved calibration fit, the computing load and if the algorithm recovers in average the original parameters. After presenting these applications, we discuss before we summarize.

2 Methodology

This section comprises three interlinked parts. We will first introduce the bi-level estimation framework; next address how to deal during calibration with constraints where both the RHS entry and resource coefficients are considered fixed, and finally introduce a modification to the objective function which reduce the step-sizes of the marginal cost curves in the LP to improve calibration.

2.1 General Setup of Bi-Level Problem

As in HOWITT (1995), we start with the following gross-margin maximization problem at farm scale:

$$\max_x \sum_j gm_j\, x_j$$
$$s.t. \sum_j a_{ij}\, x_j \le b_i \; [\alpha_i] \qquad (1)$$
$$x_j \ge 0$$

With x being the vector of non-negative decision variables indexed with j, b the constraint vector indexed with i, α the related duals, gm the vector of objective function entries (gross margins) and a the coefficients related to the constraints[1].

The PMP literature mostly discusses the case of perfect calibration against one observation of acreages and herd sizes as decision variables, treating the coefficients a as fix. Applications of EMP such as ARATA and BRITZ (2019) estimating with degrees of freedom introduce error terms on decision variables. We follow this approach in here such that observed levels might not be perfectly matched. A necessary condition for calibration of non-zero decision variables x in (1) is the balance of marginal revenues and marginal costs, potentially considering multiple observations t:

$$gm_{j,t} - \sum_i a_{ij}\alpha_{i,t} = 0 \qquad (2)$$

[1] We do not explicitly treat in the following mathematical presentation the case of equality constraints, lower and upper bounds on the decision variables or the case of free decision variables, such as an activity which captures both buying and selling of a net-put. The proposed estimator covers all these cases. The case of a double bounded variable is included in the example as crop acreages are both non-negative and partly upper bounded based on maximum shares on the total land. We also have the land balance as an equality in the constraint set of our didactic model.

This fundamental requirement is the cornerstone of PMP. Replacing the linear objective by, for instance, by a quadratic function, allows for interior solutions. Appropriate parameterization generates a strictly convex problem, such that second order conditions (SOC) hold at the calibration point. Additionally, the constraints of (1) must be feasible at the observed point.

Our bi-level program (VICENTE and CALAMA,I 1994) for calibration relates instead to a linear model and considers (certain) entries in *a*, potentially elements of *b,* the gross margins *gm* and the *x* as observed with errors. We, therefore, introduce a penalty function $g(x, gm, a, b)$ minimized under the FOCs and constraints of (1):

$$
\begin{aligned}
&\min_{x,a,gm,b,\alpha} g(x, a, gm, b) \\
&s.t \\
&\left(gm_{j,t} - \sum_i a_{ij}\alpha_{i,t}\right) x_{j,t} = 0\,, all\ j,t \\
&\left(b_i - \sum_j a_{ij}x_{j,t}\right)\alpha_{i,t} = 0, all\ i,t \\
&gm_{j,t} - \sum_i a_{ij}\alpha_{i,} \le 0, all\ i,t \\
&\sum_j a_{ij}\, x_j \le b_i \\
&x_j \ge 0 \\
&\alpha_i \ge 0
\end{aligned}
\tag{3}
$$

Bi-level problems were first introduced by Stackelberg and mainly optimize some leader strategy subject to the followers' optimal answer. Here, they are applied to an estimation framework where the inner problem comprises the constraints and FOC of model (1). The outer problem is the estimator which "proposes" parameters to the inner one. It "responds" by returning the optimal primal and dual solution *x, α* at these parameter values or by reporting the infeasibilities. This process in encapsulated in a gradient based solver. The applications of EMP by JANSSON and HECKELEI (2011) or ARATA and BRITZ (2019) are formally equal to (3), but relate to problems with a quadratic objective.

Our application differs from these examples by considering multiple constraints as inequalities. More important, there is no second-differentiable production (or cost respectively profit) function which guarantees a unique solution to (3) and/or helps to reduce non-convexities in the solution space. Equally, the number of inequalities in our model is large compared to the number of decision variables, a typical feature of bio-economic models. Finally, we also consider the case of integer variables.

In opposite to the linear model (1) to optimize, the bi-level calibration problem (3) comprises cubic equality and quadratic in-equality constraints: the parameters *a,* the decisions variables *x* and the duals α are estimated simultaneously and their products occur in its first three constraints of (3). The first two cubic expressions are not strictly convex. They express FOC in the Karush-Kuhn-Tucker (KKT) form, due to non-negativity conditions on *x* and the presence of inequality constraints; related duals α can be zero. These non-convexities render the problem numerically demanding. We present therefore an algorithm to improve solving such bi-level programming approaches.

We assess below applications with negative degrees of freedom where more parameters are estimated than elements of *x* entering the penalty function, such that perfect calibration is likely possible.[2] Solving simultaneously for multiple years can imply positive degrees of freedom, instead. The bi-level problem (3) captures hence the continuum between (perfect) calibration and an estimation of parameters of the Leontief production function in (1) under the assumption of profit maximizing behaviour.

2.2 Data Balancing

The PMP literature rarely discusses cases where (1) is infeasible at the calibration point, probably as PMP based models typically comprise few constraints. Many PMP based models comprise, besides a land balance, only restrictions relating to policy instruments such as set-aside obligations which act as a (direct) binding constraints to the production mix (cf. FFSIM: LOUHICHI et al., 2010; IFM-CAP: LOUHICHI et al., 2015). In detailed bio-economic models, infeasibilities are more likely to occur during calibration. (3) comprises the constraints of the model used for simulation (1), such that the bi-level estimator will also (try to) remove any

[2] As we will discuss later, degenerate solutions are commonly found and will mean that perfect calibration in the strict sense is not possible. See next section.

infeasibilities implied by the coefficients a and the RHS vector b at the observed point x.

If not all a and b are subject to calibration, or bounds on them present, feasibility and thus perfect calibration cannot be guaranteed. We, therefore, minimize first squared differences between the calibration targets $\widehat{x_{j,t}}$ and a feasible solution $x_{j,t}$ in a separate data balancing step (4). This sub-model comprises solely the constraints comprising fixed coefficients not subject to estimation, in here the land balance and maximal crop shares. Optimizing (4) provides a new solution $x_{j,t}$ where perfect calibration is at least technically feasible, and shows the impact of fixing some coefficients. In real world models, (4) might comprise constraints relating to policy instruments where resource coefficients and RHS entries are fixed and given by the law book.

$$\begin{aligned} &\min_x g(x) = \sum_{j,t} \left(x_{j,t} - \widehat{x_{j,t}}\right)^2 \\ &\bar{b_\iota} - \sum_j \overline{a_{\iota j}} x_{j,t} \geq 0, some\ i, all\ t \\ &x_j \geq 0 \end{aligned} \tag{4}$$

Using only constraints with fixed coefficients in (4) results in linear constrains. This allows including integer variables as modern MIP solvers work efficiently for linearly constrained problems with a quadratic objective. Subsequent calibration steps should be judged against the solution of (4), and not against the original targets $\widehat{x_{j,t}}$. (4) can be dropped if all coefficients a and b are subject to calibration and their bounds do not prevent perfect calibration, or if sub-perfect calibration is accepted. The link of (4) to the bi-level problem (3) is threefold. First, the fixed coefficients in (4) will also be fixed in (3) and second, we will judge in the following the fit achieved by (3) against the solution of (4), and third, (4) provides a preliminary estimate for integer variables.

2.3 Avoiding Degenerate Solutions

For any vector of shadow prices α, there exists a vector of gross margins gm balancing the exhaustion conditions (2) for the non-zero elements of x. If the number of non-zero x exceed the number of binding constraints, a set of linear dependent solutions will result. In this case, some decision variables can be shifted away from the calibration point(s) and others adjusted to keep the constraints feasible without changing the objective value. This is one reason why non-linearities are introduced in PMP.

Here, we borrow the idea of PMP to introduce some costs increasing in x, using a step-wise linearization approach as proposed by SCHMID and SINABELL (2005). We introduce gross margin depression effects from more specialized programs, by reducing the gross margin of any crop stepwise by not more in total than 1% of their expected mean; from a zero crop share to the maximal share allowed. These small gross margin changes should hardly impact the allocative response of the model and could in real-world applications be based on empirical evidence, e.g. considering preceding crop effects. Let $gmRed_{j,t,gmd}$ [ha] measure the amount by which crop j in year t exceeds an entry in *critShare,* a matrix of given crop specific shares at which a further gross margin depressions *gmd* occur. b_{land} depicts the available land [ha] and *gmDepr* [€/ha] the changes in gross margins related to *critShare* such that the farm's total gross margin gm^* to maximize now becomes:

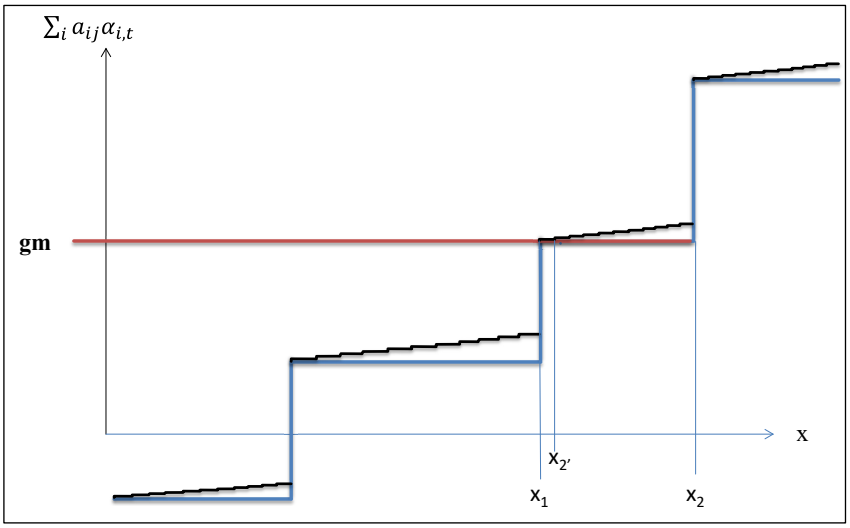
$$\begin{aligned} &gmRed_{j,t,gmd} \\ &\geq x_j - \overline{critShare}_{j,t,gmd} \bar{b}_{land} \\ &gm^* \\ &= \sum_{j,t} \left(x_{j,t} \overline{gm_{j,t}} \right. \\ &\left. - \sum_{gmd} gmRed_{j,t,gmd} \overline{gmDepr_j} \right) \end{aligned} \tag{5}$$

The terms $x_{j,t} gm_{j,t}$ in the new objective function shown in the second line capture the gross margin changes of a crop between a zero share and the first element of *critShare*, the second term stepwise decreases each crop's gross margin as its share grows. Assume 100 critical shares 1%, 2%....100% and a maximal gross margin change of 1% of 500 €/ha, i.e. 5 € /ha in total. This results in a uniform decrease in gross margins of 5/100 = 0.05 €/ha in each depression step and provides a stepwise linearization of a quadratic relation between the gross margin of a crop *gm* and its acreage x. By using non-uniform gross margin decrements, any other convex relation might be recovered. We opt for an approach which increases the step-width up to the maximal allowed crop share. The resulting larger gross-margin changes at smaller crop shares are deemed favourable in calibration as they reduce relative errors.

In Figure 1, the blue curve indicates the marginal costs before introducing the terms $\sum_{gmd} gmRed_{j,t,gmd} \overline{gmDepr}_{j,gmd}$ in the objective, while the black one depicts them after their integration. The modification to the objective function reduces the step-width of the stepwise marginal cost curve of the LP. This dampens the unwanted impact of a potentially degenerate solution resulting from calibration where the primal solution can move between neighbouring points of the solution space without changing the objective. Such a case is depicted in Figure 1 where at the given gross margin *gm*, the decision variable can move between the points x_1 and x_2, reduced to x_1 to $x_{2'}$ by the modification. This stabilizes the bi-level estimation framework as moving variables by more than the current step-width according to (5) will imply some small changes in the overall gross margin (steps on the black marginal cost curve in Figure 1). We consider fifty steps and let the step width at which changes occur increase in the crop share with an exponent of 1.5, instead of using linear increments. This renders is likely that the relevant flat ranges depicting degenerate solutions after the modification become smaller (x_1 to $x_{2'}$ in Figure 1). Due to the non-linearity, this range will be determined by the step width of the crop with the smallest crop share involved in the degeneracy. This is likely smaller than the average step width under the linear solution (100%/50) as two crops with more than a 50% crop share can never occur in a degenerate solution.

When referring in the following to (1) and the related bi-level problem (3), these problems comprise the (updated) equations from (5), i.e. the additional inequalities defining *gmRed* and the updated objective function and its FOC.

Figure 1. Modified marginal cost curve



Source: authors

3 Technical Implementation and Restart Heuristics

We describe in the following three key elements of implementing the bi-level estimation. First, the use of a specific package in GAMS which eases its formulation and solution, second, how to detect primal degenerate solutions, and third, how to improve in cases the solver only finds a local optimum in the non-convex solution space, by systematically restarting the solver with different start values.

3.1 Implementation and Solving of the Bi-Level Problem

Our didactic example is encoded in GAMS (General Algebraic Modelling System), a set-driven Algebraic Modelling Language easing the translation from mathematical notation into computer code. GAMS provides transparent interfaces to solvers to efficiently optimize the LP or MIP (1) and the bi-level NLP problem (3). Program control features such as loop and if-else blocks and in-built functions allow us to perform the Monte-Carlo analysis presented below. The code is found in the annex (bilevel_calmip.gms).

In order to generate problem (3) from an existing model (1), the FOC would need to be coded and equations defined where constraints are multiplied with their duals. This is cumbersome and error-prone for detailed bio-economic farm-scale models comprising hundredth of different variables and equations. Furthermore, gradient based solvers for NLP problems will easily fail on the not strictly convex KKTs in (3). This is why specialized solvers are used for Mixed-Complementary Problems (MCP). However, such MCP problems do not comprise an objective function. We therefore draw on the Extended Mathematical Programming package (EMP, FERRIS et al., 2009) and the NLPEC solver (FERRIS and GAMS DEVELOPMENT, 2009). We only need to define the simulation model (1) and the penalty function $g(x, gm, a, b)$ in GAMS, as the EMP package allows, inter-alia, automatic formulation of the FOCs of a bi-level program. The NLPEC solver then provides smooth approximations of the KKT conditions to overcome convexity issues. It delegates the generated bi-level program to a multi-purpose NLP solver.

Generally, the bi-level problem should comprise all potentially binding constraints. For sake of simplicity, all constraints can be added, including cases such as pure reporting identities known to never carry a dual value. So far, not many applications of this package to economic problems are found in literature (cf. BRITZ et al., 2013; KUHN et al., 2016; ARATA and BRITZ, 2019).

Before solving the bi-level problem, we optimize model (1) at current parameter values. This provides a feasible starting solution to (3) as all FOCs of the lower model are satisfied and no constraints are violated. From there, the solver starts its search towards a solution close to the calibration targets by updating parameters. We use the so-called penalty formulation offered by the EMP package which moves the KKT conditions into the objective function of the bi-level program. Accordingly, only constraints including bounds on decision variables of (1) enter (3) as constraints. We solve the problem twice, first with a weight of 1/100 for the KKT conditions in the objective (*initMu* in the NLPEC steering file). This low weight for KKT violations allows the solver to update parameters quite easily. However, this implies that the resulting solution only represents a primal feasible solution to the LP with considerable dual infeasibilities, i.e. the resulting solution is sub-optimal. In a second final solve, the weight for KKT violations is increased to 1.E+6 (*finalMu* in the NLPEC steering file) to remove dual infeasibilities. CONOPT4 finds a local minimum to the bi-level problem typically in a few seconds as long as the number of observations is small, benefitting from parallelism.

3.2 Detecting Degenerate Solutions and Improving on Local Optima

Problem (3) is not convex as the SOC are zero. It allows for (and will in many cases result in) degenerate solutions rooting in linear dependencies. Restarting the solver on the simulation model (1) at estimated parameters will likely simply reproduce the primal solution of (3), even if it is degenerate. Therefore, after solving (3), we reset all decision variables to zero and solve the LP at the estimated parameters with a different solver to check for deviations. In case of integers being present, they will be unfixed in this solve which can provoke additionally differences between the optimum of the bi-level problem and the subsequent solve of (1).

Finding a global optimum for a bi-level programming approach where the inner problem comprises inequalities and bounds is computationally challenging. Improved calibration of the model typically requires switching the slack status of multiple inequality constraints. A gradient based solver will typically not consider changes in the RHS or coefficients in currently non-binding constraints, as they have no impact on the duals or the feasibility status. It will, therefore, tend to find a local optimum only, depending on its start values. To improve here, we need an outer loop which offers different start points to the gradient based solver (even if no integers are present).

With a sufficiently small parameter space, systematic search algorithms such as a grid search might be used for this. As already smaller changes in parameters can have a larger impact on the model's fit, a quite fine-grained grid would be needed. Already three grid points such as [lower bound, mode, upper bound] for each of the twenty parameters in our model would mean 3^20 tries which is computationally impossible. We draw here instead on the algorithm proposed by SCHÄFER and BRITZ (2017) which construct restart points by perturbing stochastically the current best parameters (see also Figure 3). [3]

The outer-loop requires at least a maximum number of re-starts of the bi-level problem as a stopping criterion, here chosen as hundred to test many models in a Monte-Carlo analysis. In applications to one actual model, only, a higher number might be appropriate. Besides restricting the maximum number of starting points, one might stop testing further restart points once a parameter set offers a satisfactory fit, here defined based on the deviations from the calibration targets, and not considering deviations from expected parameters values:

$$fit = \frac{1}{T * J} \sum_{j,t} \left(x_{j,t} - \widehat{x_{j,t}}\right)^2 \qquad (6)$$

[3] We repeatedly switch between five variants of the perturbation. The first two perturb based on the current best set of parameters, only, multiplying each parameter with $N(1; 0.05)$ and $N(1; 0.10)$, respectively. The third one multiplies it with $N(1; 0.10)$ and adds $N(1; 0.01)$ times the a-priori mean of all parameters. Similarly, the fourth one multiplies the current best parameters with $N(1; 0.50)$ and adds $N(1; 0.10)$ times the a-priori mean to others. The last one only uses the expected mode of the parameters, adding $u(-0.5; 2.5)$. These choices are clearly arbitrary and the outcome of some testing. Still, as they are defined relative to the expected parameter modes and current best parameter values, they can be used with other problems as well.

The fit is not measured based on the solution of (3) returned by the solver. Instead, we reset all x to zero and, potentially, unfix integers and solve the primal model (1) at the parameter estimates, to detect potential degenerate solutions.

Repeated restarts until a very accurate threshold in (6) is reached will improve in average over many calibration exercises the fit at the cost of higher run time and vice versa. We opt here for a compromise which increases the threshold of what is considered a sufficient fit the more restart points where already tested. In the didactic application below, the fit relates to average squared deviations in ha, with acreages adding up to 100. We start with a minimal fit of zero in the first trial, linearly increased to two at maximum of 100 trials. Starting with a very low threshold will trigger restarts of the bi-level estimator even if the accuracy of early trials is already quite high to allow for cheap further improvements. We consider the chosen maximal threshold of two as sufficiently close to the "true" observations after one hundred trials; the typical fit achieved is far better as discussed below, as long as the number of observations is small. Increasing the threshold slightly after each restart also reflects that perfect calibration with multiple observations is often impossible. Furthermore, as discussed in section 2.3 above, the step sizes of the gross margin changes in (5) determine ultimately the possible calibration quality.

Incorporating integer variables directly in the bi-level estimation framework is not possible with our solver, preventing a generic approach for MIP problems. If coefficients relating to integer variables are treated as fixed and given, integers variables can be fixed in the bi-level estimator. They become free variables in the subsequent solve at currently drawn parameters to assess the fit. In our tests, the algorithm could in average considerably improve the calibration compared to the "true" parameters with integer variables[4] present, as long as the number of observations was small. That is surprising as, once integers are present, not only the FOC must hold at the current parameter choice, but the resulting objective must also be higher than any other integer solution. This condition cannot be tested in an estimation framework for larger number of integers.

4 An Example Application

The example application uses a clearly defined data generation process and not empirical observations, such as found in Farm Accountancy Data Network. This allows to judge the properties of the bi-level estimation process, for instance, with regard to recover "true" para-meters, which are unknown when using empirical data. Instead, we construct rather simple LP and MIP models for which we draw parameters in a Monte-Carlo analysis, perturb their optimal solution with white noise and then let the bi-level estimator find parameters which calibrate them.

4.1 Model Setup

Our didactic model (see Table 1) covers five crops and comprises elements related to crop production found in most bio-physical models: a land balance, maximal cropping shares, and annual and monthly labour constraints. The version comprising integers add machinery requirements and related investments. While land and yearly labour endowments as well as maximal crop shares are fixed, we generate variants based on Monte-Carlo analysis where other endowments and model coefficients differ.

Specifically, we consider different labour endowment available in three peak months as variying constraints. Each peak month has a labour endowment of 50% above the 1,000 hours available annually divided by twelve months, perturbed for the variants by a uniform distribution of $u(0.9;1.1)$. Revenues in € per ha for each crop and year are drawn stochastically from a uniform distribution $u(1{,}000;1{,}200)$ and costs in € per ha for each crop from $u(400;500)$. We infer from there labour needs per hour by making an assumption how much of the total gross margin of the farm remunerates land. The remainder of the total gross margin is used to estimate a shadow price per labour hour, and based on this, to derive an estimate of total labour needs per ha from revenue exhaustion for each crop. The total annual labour needs for each crop serve as a basis to allocate labour need shares to the three peak months, multiplied with $u(0.75;1.25)$. For each observed year, we have hence ten potentially binding constraints (land, four labour constraints, five maximal cropping shares) and five decision variables.

[4] The GAMS code comprises the necessary code to test the discussed extension of the model with integer variables and the changes in calibration set-up.

Table 1. Characteristics of didactic models subject to Monte-Carlo analysis

Model block	Elements	Details
Decision variables and objective function, LP	Five crop acreages for 1 to 10 years	Revenues differ by year (fixed), variable cost time independent (MC)
Decision variables and objective function, MIP	As above, integer based investments into different machines by type and size, once for all years	Machinery costs and yearly operation capacity (fixed)
Constraints	Land balance	RHS: 100 ha (fixed)
	Total annual labour	RHS: 1,000 hours (fixed); crop specific labour needs (MC)
	Labour in three peak months	RHS: 1,000 hours / 12 times * 1.5 (+MC); crop specific labour needs (MC)
	Maximal crop shares	Fixed
	Machinery needs (only MIP)	Fixed

Note: Elements indicated as fixed are not subject to calibration, MC: element is subject to Monte-Carlo analysis to test calibration of different variants of the same structural model.
Source: authors

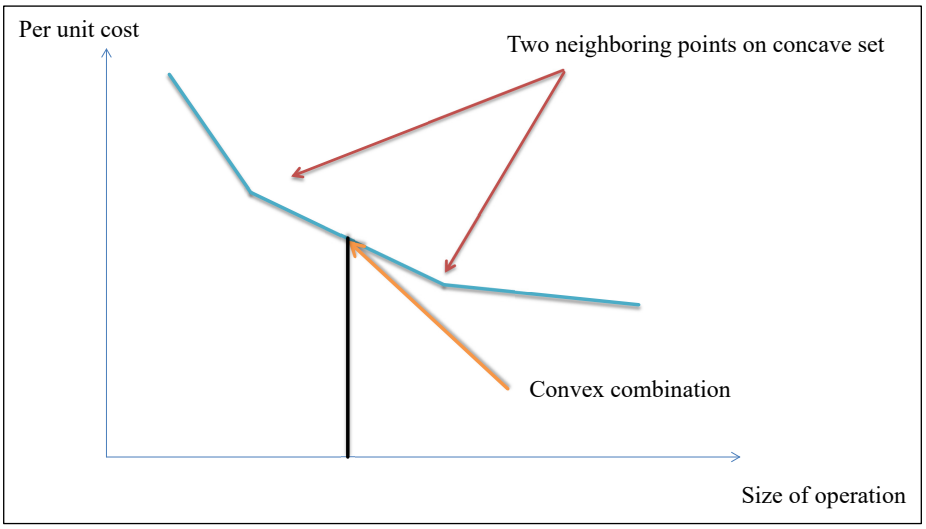
In the model version with integer variables, we introduce machinery requirements covered by investments taken for the whole period. We consider five machine types of different sizes $[10; 50; 100]$, referring to the acreage in ha which can be operated per year. Related annual investment costs are $[1{,}000; 4{,}000; 6{,}000]$ €. This implies costs of $[100; 80; 60]$ €/ha and year at full capacity use, i.e. increasing returns-to-scale. For each machinery type, a convex combination between neighbouring points of the concave set is endogenously chosen, where the chosen points are depicted by integer variables (see Figure 2). Machinery requirements at farm level are defined as the total acreage covered by the crops needing the machine, in average over the years. Further details can be found in the GAMS code in the annex and in the supplementary material.

Equation (7) below depicts the penalty function $g(x, gm, a, b)$ chosen for (3). Entries of crops into the land constraint are unity and not subject to calibration. The same holds for the maximal crop shares. Labour demands and the per hectare costs c are subject to calibration, they are considered time-invariant, while we treat the RHS vector b as given.

We introduce as the first term in the penalty function absolute squared differences between the simulated x_j and observed acreages $\hat{x}_j$, identical to (6). Second, we add relative squared differences between the estimated per ha cost c_j and the "true" parameters c_j in the penalty function, and finally, squared relative differences for the resource coefficients subject to calibration:

$$\min_{x,c,a,\alpha} g(x,c,a) = w_x \sum_{j,t} \left(x_{j,t} - \widehat{x_{j,t}}\right)^2 + w_c \sum_{j} \left(\frac{c_j - \widehat{c_j}}{\widehat{c_j}}\right)^2 + w_A \sum_{i,j} \left(\frac{a_{ij} - \widehat{a_{ij}}}{\widehat{a_{ij}}}\right)^2 \quad (7)$$

Figure 2. Convex combination over a concave set of investment possibilities to depict returns-to-scale



Source: authors

The weights w_x for definition in x are as in (6); w_c referring to the per ha costs c is equal to the number of crops, and the weights for the resource coefficients subject to calibration w_A are equal to the number of constraints times the number crops. In a real world application, differentiated weights across types of decision variables are likely, such as for crops and herd sizes. The use of relative squared deviations for the parameters is motivated by the fact there is no common unit across resource and objective function coeffi-

cients, such that penalizing absolute changes would make limited sense. This decision and the choice of the weights w_x, w_c and w_A will affect the estimates, especially the trade-off between model fit and deviations from the a-priori information on (certain) parameters. Results in here (see section 4.3) show small estimated errors on the here known coefficients in combination with a good fit. This gives some confidence that the chosen weights were appropriate. For empirical applications, a sensitivity analysis with the weights might be warranted.

In order to test the calibration process, we first optimize (1) at the "true" parameters (partly randomly drawn). Next we add white noise error terms with differing $\sigma^2 = [2,5,10]$ to the optimal solution x_{true} to generate random calibration targets:

$$\widehat{x_{j,t}} = x_{j,t}^{true} + n(0,\sigma^2) \tag{8}$$

This implies that allocative decisions are observed with errors. The follow-up data balancing step ensures that calibration targets do not violate the maximal crop shares, the land balance and non-negative conditions; constraints with coefficients not subject to calibration. This step minimizes the first term of (7), targeting the outcome of (8):

$$\begin{aligned} \min_x g(x) &= \sum_{j,t} \left(x_{j,t} - \widehat{x_{j,t}}\right)^2 \\ s.t. & \\ \sum_j x_{j,t} &= b_{land} \\ 0 \le x_{j,t} &\le maxShare_j \end{aligned} \tag{9}$$

The data generation process resulting from (8) and (9) cannot generate $x_{j,t}^{true}$ in average where some crops will be frequently zero at the optimum. Adding errors terms in (8) to zero observations and truncating at zero to ensure non-negativity of the observed in (9) will bring such zero observations in average over the draws into the solution. The land balance will then imply downward bias for non-zero observations.

If the model is solved for multiple years, revenues *r* per hectare change stochastically across years, while all estimated coefficients are time invariant. If perfect calibration is possible will depend on the number of time points, as discussed in the next section.

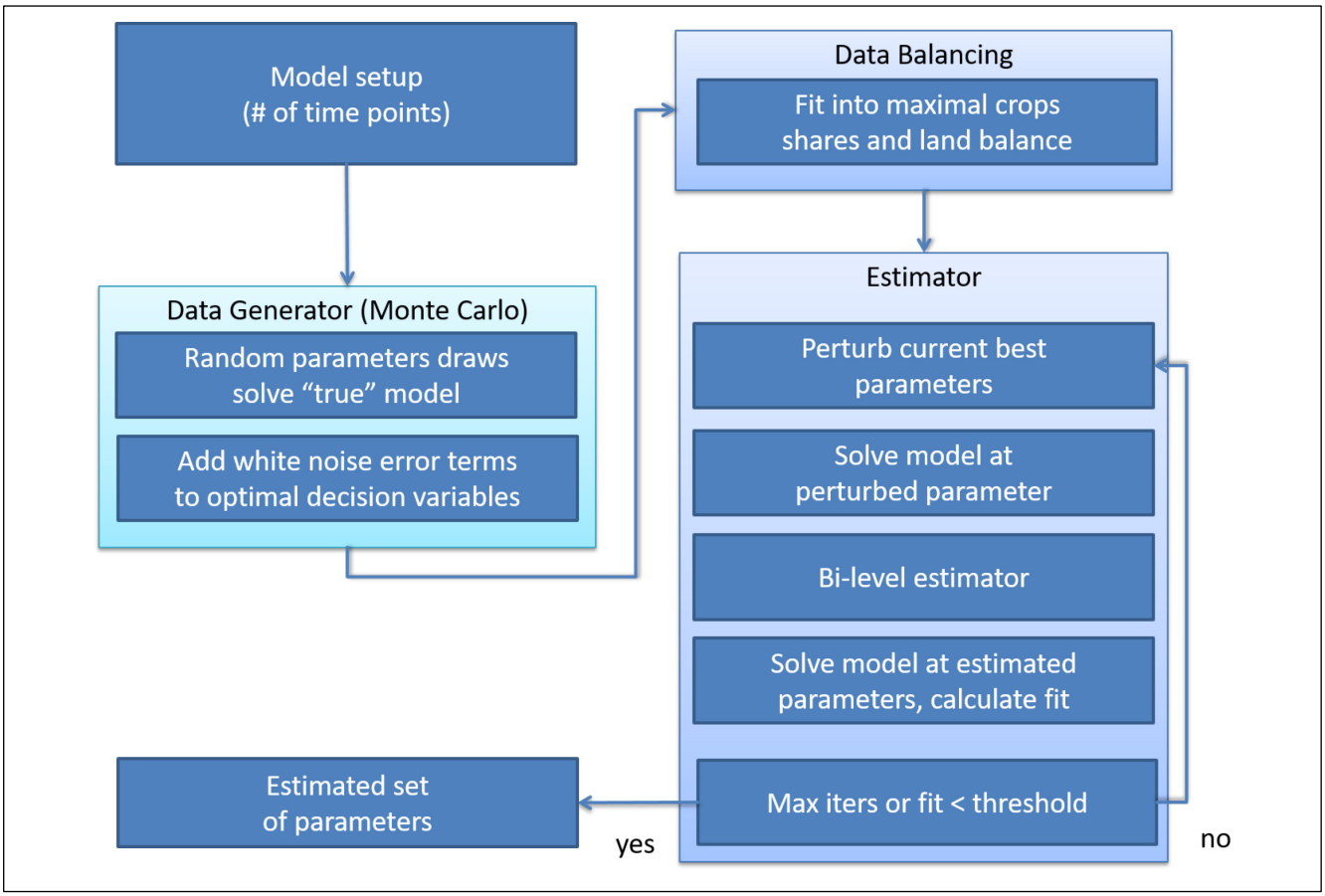
There is a trade-off in (7) between the calibration fit and staying close to the mode of the parameters. Setting the weights for parameters to zero will imply uniformed priors. That is, however, dangerous as e.g. the overall farm gross margin is not controlled during estimation. If the cost entries *c* for all crops are changed by the same absolute amount, the dual of the land balance as an equality constraint will change accordingly. There are hence likely linear dependencies in the model. This problem can be partly avoided by removing the land balance and its dual value from the lower level and move the land balance instead in the upper level in (3). After the solution, the costs of all crops can be reduced by an assumed land rent. A similar approach was used by cf. KANELLOPOULOS et al. (2010) in a PMP model. We consider here the more usual case where a priori information on the parameters is used, and duals are not explicitly controlled.

4.2 Set-up of the Tests and Assessing the Performance of the Estimator

We set-up the model such that it comprises one or multiple years (see also Figure 3). For each number of years, we generate fifty model variants by randomly drawing parameters, i.e. yearly revenues, time invariant costs and four labour requirements for each crop. Maximum crop shares and farm endowments besides labour availability in peak months remain unchanged. For each variant we first determine the optimal solution at this given parameters. From there, we generate twenty different calibration targets by adding white noise error terms to the optimal solution for all crops, including unobserved ones. We hence consider the LP as the true data generation process. We explore to what extend the algorithm recovers in average the true parameters, assessing in parallel its calibration performance. For each number of years and the different variances of the error terms considered, this implies 1,000 draws (fifty different sets of parameters times twenty different error terms). For each draw, the search algorithm will test up to hundred perturbations of the parameters and related restarts of the solver on the bi-level problem. As discussed above, the required fit to stop the search process increases linearly from zero for the first try to two [ha²].

For each instance and draw, we solve the data balancing problem (4) and use its solution as the calibration target in (7). We next (a) perturb the current best solution of the parameters, (b) solve the primal model (1) at fixed resulting parameter values, (c) unfix the parameter and solve (3) (at fixed integers in case of the MIP model), (d) fix the parameters to the solution of (3), set decision variables to zero, unfix the integers in case of MIP, and solve problem (1). The results from (1) determine the fit. If the fit is above the threshold and

Figure 3. Overview on methodology



Source: authors

the maximal number of restarts not yet reached, we repeat (a)-(d).

The twenty randomly drawn labour requirements and the five cost entries in the objective are the parameters subject to calibration. Jointly with the number of observations and crops, they determine the degrees of freedom. A model comprising one year, only, suggests an undetermined model with twenty free parameters (25 parameters minus 5 FOC conditions). Bounds on the parameter, in-equalities and non-linearities in (3) exclude an exact definition of the degrees of freedom. Already with two years, (almost) perfect calibration was not always possible. In this case, we face 2x5=10 FOC conditions. We lose for each of the potentially four binding labour constraints in each year one degree of freedom, as one of the crop specific coefficients is depending on the others. Along with bounds on parameters, this explains cases where perfect calibration is impossible already with two years. For such cases, the algorithm acts as Bayesian estimator where the structure of the model and the observations jointly determine the posteriori of the parameters and the error terms. Minimizing squared relative differences of the parameters in combination with some weights to express a-priori information is a rather intuitive approach. Formally, under normal distributed errors, the approach maximizes the log-likelihood of the posterior density (HECKELEI et al., 2008) which further motivates its use as the penalty function. The quadratic function with its linear derivatives is also a good choice for gradient based solvers, but alternatives such as a (cross)-entropy approach could be used as well.

We assess the algorithm firstly by its ability to calibrate the model sufficiently accurate based on the average fit. Its efficiency is assessed by the number of required restarts which strongly impact computing time.

4.3 Results

We test the case of one year (= one observation) with different standard deviations of the white noise error added to the "true" model solution. With one observation, the coefficient of variance for the estimated parameters is in the one to five percent range, depending on the variance of the error term, while mean deviations from the true parameters are below one percent (see Table 2). Deviations are expected due to truncation in the

Table 2. Mean error, coefficient of variance of error and mean relative error of estimated parameters, no integers

	$N(0;2)$	$N(0;5)$	$N(0;10)$
Per ha costs	-0.1282 (0.015,-0.0001)	-0.0967 (0.016,-0.0001)	-0.0769 (0.014,-0.0001)
Labour per ha, total	+0.0004 (0.010,+0.0002)	-0.0086 (0.012,-0.0007)	-0.0071 (0.012, +0.0009)
Labour per ha, July	-0.0019 (0.027,-0,0002)	-0.0071 (0.031,-0.0042)	-0.0081 (0.033, -0.0048)
Labour per ha, August	0.0057 (0.025,+0.0056)	+0.0044 (0.028,+0.0045)	+0.0071 (0.031, +0.0066)
Labour per ha, September	0.0008 (0.020,+0.0016)	+0.0019 (0.030,+0.0030)	+0.0019 (0.030, +0.0030)

Note: parameter statistics are expressed as averages over crops; coefficient of variance (first) and mean relative mean error (second) in brackets. Column headings refer to the white error noise added to the optimal solution of the model variant.
Source: authors` calculations

Table 3. Quality of fit and required restarts for the case of one observation, no integers

	$N(0;2)$	$N(0;5)$	$N(0;10)$
Fit of simulation model	0.022 (0-0.19)	0.026 (0-0.33)	0.027 (0-0.26)
Fit of bi-level estimator	0.0027 (0-0.19)	0.0013 (0-0.15)	0.0004 (0-0.24)
Error term	2.02 (0-12.07)	12.95 (0– 94.00)	48.80 (0-299)
Fit improvement	-94.65%	-97.73%	-98.65%
# of starts	2.04 (1-13)	2.27 (1-16)	2.57 (1-18)

Note: (1) Fit measured as squared deviations between calibration target and solution in ha, divided by number of observation, i.e. five, reflecting the number of crops. (2) First number in each cell is the mean; min and max in brackets. (3) The error term measures the average squared deviation of the calibration target from the uncalibrated model solution. (4) Fit improvement: relation between fit of calibrated model and uncalibrated one as expressed by the error term.
Source: authors` calculations

data generation process discussed above, which also implies bias for estimated parameters.

Against that background, the almost perfect fit might be astonishing, even more so the low coefficients of variance. Equally, the size of the error term has a quite limited impact on the statistics. This might firstly reflect quite limited stability ranges of the drawn solutions, i.e. relative small changes in coefficients of the "true" simulation model lead to larger changes in the primal solution. Secondly, the maximal crop shares determine to some extent the solution. For a crop bouncing against its upper share limit in the "true" case, a positive perturbation will mean that data balancing based on (5) will pull them to, or close to the upper limit. We might hence expect similar effects in models with a richer constraint set narrowing down the solution space.

The average fit of the calibration with one observation (see Table 3) above does almost not respond to the average size of white noise errors error added, and is very small with just 0.022 to 0.027. The calibration fit which can be achieved reflects, as discussed above, the step-width of the gross margin depression mechanism. The average fit of the bi-level estimator suggests that it is almost always possible to find a parameter sets for which the FOC hold at the calibration target. Increasing the variance of the error term has a very moderate impact on the performance of the estimator. There are no considerable differences in the fit from a model application perspective, i.e. with regard to the fit of the simulation model, or with regard to required restarts of the model, when the average distance between the "true" model and the calibration target increases considerably.

As expected, introducing integer variables renders calibration harder (see Table 4): it drives up the required average restarts substantially and reduces considerably the fit. This also reflects that the threshold increases dynamically with the number of restarts, i.e. if the algorithm does not find a good fit after a few restarts, it will also consider a lower fit as sufficient to stop further searches.

Overall, the results suggest that an acceptable calibration might also be possible in case of integer variables. Introducing integers *reduces* in here the average error term, but worsens the fit of the estimation step. Equally, integer variables drive up the number of required restarts. The reason for the in average smaller error term is the returns-to-scale effect which favours more specialized solutions, where crops hit more often their maximal cropping shares or don't enter the optimal solution. The data balancing step might be pushed

Table 4. Quality of fit and required restarts for the case of one observation, integers included

	$N(0; 2)$	$N(0; 5)$	$N(0; 10)$
Fit of simulation model	0.227 (0-6.64)	0.650 (0-32.97)	1.615 (0-61.95)
Fit of bi-level estimator	5.526 (0-1105)	4.757 (0-1559)	1.254 (0-179)
Error term	1.958 (0-12.11)	12.622 (0–98.23)	46.857 (0-349)
Fit improvement	-81.20%	-93.56%	-95.03%
# of starts	14.51 (1-100)	18.46 (1-100)	23.22 (1-100)

Note: for explanations, see Table 3.
Source: authors` calculations

Table 5. Quality of fit and required restarts for different number of observations, no integers

# of obs.	Fit of simulation model	Fit of bi-level estimator	Error term	Fit improvement	# of starts
1	0.022 (0-0.192)	0.0027 (0-0.19)	2.02 (0-12.07)	-94.65%	2.04 (1-13)
2	0.048 (0-6.06)	0.020 (0-6.06)	2.17 (0.01-8.62)	-96.90%	2.97 (1-100)
3	0.114 (0-3.33)	0.076 (0-2.78)	2.17 (0.06-7.75)	-94.07%	6.03 (1-100)
5	0.259 (0-1.60)	0.210 (0-1.58)	2.08 (0.19-6.07)	-86.35%	11.73 (1-67)
10	0.792 (0.15-1.96)	0.760 (0.12-1.93)	2.05 (0.55-3.94)	-60.98%	30.76 (1-75)

Note: for explanations, see Table 3.
Source: authors` calculations

Table 6. Quality of fit and required restarts for different number of observations, with integers

# of obs.	Fit of simulation model	Fit of bi-level estimator	Error term	Fit improvement	# of starts
1	0.227 (0-6.64)	5.526 (0-1105)	1.958 (0-12.11)	-81.199%	14.51 (1-100)
2	0.470 (0-8.01)	3.267 (0-846)	2.092 (0.01-10.76)	-74.85%	26.72 (1-100)
3	0.579 (0-6.66)	6.442 (0-1980)	2.098 (0.01-7.78)	-70.50%	29.38 (1-100)
5*	0.916 (0.002-4.83)	1.250 (0-187)	2.079 (0.18-5.69)	-56.32%	38.50 (1-100)
10**	1.057 (0.20-3.40)	1.001 (0.13-3.62)	1.953 (0.67-3.53))	-47.78%	40.75 (10-100)

Note: for explanations, see Table 3. * based on 33x20 experiments. ** based on 5x20 experiments.
Source: authors` calculations

in some instances towards the simulated solution. The reduced overall fit is expected as the parameters related to the machinery requirements restrictions which are linked to the integers are not under control of the estimator.

We test the case of multiple observations (i.e. a simultaneous solution for multiple years) only for the case of $N(0; 2)$, for results see Table 5, which comprises for comparison the single observation case. The average number of restarts with two observations increases from around 2.42 to around 2.97, while the fit worsens only slightly to 0.048. This might still be considered a perfect calibration in terms of average squared differences in ha for a farm with 100 ha. A drop in the calibration performance with multiple observations is expected. In the case of one observation, only, the per ha crop costs alone can be used to solve for the FOC of type (2). Calibration in case of two or more observations requires updates to other coefficients as well. Parameter updates will change the primal solution if they relate to binding constraints, only, and will typically impact the fit of all years simultaneously. In case of two observations, there was one draw out of the thousand with a larger error term where the algorithm could not find any solution improving the fit. In case of three observations, larger calibration errors are quite common reflecting the fact that degrees of freedom seem regularly exhausted.

With higher number of time points, the problem is no longer undetermined for many of the draws. That reduces the average fit. Due to the dynamic thresholds in the outer loop of the algorithm, this also implies in average more starts of the solver. Still, for up to five observations, the fit of the calibrated model is substantially improved compared to the uncalibrated one.

The impact of introducing integer variables on the fit and more so on the required restart is much stronger as seen from Table 6. The integers with their related constraints restrict the solution space considerably further compared to the LP model above, and drive the

model towards more specialized solutions. This implies that the changes to the parameters subject to estimation are less likely to allow fitting to given calibration targets, which differ from the model solution at the "true" parameters. This optimal solution at the true parameters clearly also reflects the integers and related constraints, but the related coefficient entries are here not considered uncertain. Furthermore, integers are fixed to their optimal values at the currently drawn parameter values during estimation such that the estimator cannot consider that other integer values might become optimal if parameters are updated. This becomes only visible in the subsequent solve of (1).

Already with five observations, finding solutions with a good fit with integer present takes very long.[5] In many cases, even hundred restarts could not deliver a fit better than the maximal considered threshold of two.

Our model is quite small such that one might expect a bi-level problem with a few variables and equations, only. Due to the gross margin depression mechanism, the simulation model comprises, however, for each time point and crop 50 non-negative variables and inequalities. In the bi-level program, that implies 200 endogenous variables: the primal variables and related duals as well as the slacks in the equations and their duals. For five crops and five periods, that already implies 200x5x5 = 5,000 variables. Due to complementarity conditions, the EMP package will automatically remove some of them from the model. Still, the resulting bi-level estimation problem for five time points comprises slightly more than 2,500 equations and 5,000 variables.

5 Discussion

Compared to PMP, the proposed calibration method has a different focus. In PMP, the coefficient matrix is usually taken as fix and given, while non-linearities are introduced to capture "unobserved" costs or revenues. We consider instead objective function entries and (some) resource coefficients as observed with errors, minimized during calibration, without adding new elements to the model. If the model subject to calibration likely misses constraints faced by the farmer and/or in other cases of (other) assumed unobserved costs or revenues, a PMP approach might be more appropriate. Another difference to PMP is that we refrain from structural model changes which strongly impact its allocative response. The proposed gross margin depression mechanism is deliberately parameterized such that it "just" allows for "interior" solutions while largely maintaining the step-wise marginal cost curves of the original LP. PMP instead overwrites the solution behaviour of a LP with a low number of constraints. Closed-form solutions (c.f MEREL and BUCARAM, 2010) are now available to calibrate different types of PMP models against (certain sets of) price elasticities. In our framework, price elasticities could be used to introduce observations with updated prices and decision variables.

Using our approach with a higher number of observations is a special case of EMP where profit-maximal parameters of a Leontief production function are estimated. Compared to e.g. ARATA and BRITZ (2019) the estimator faces in our examples more inequality constraints and considers integers. Resulting non-convexities in the solution space of the bi-level problem require restart heuristics to find better local optima. This cannot guarantee a global optimum, but is found to improve considerably the fit. Besides a maximal number of restarts, we used a minimal fit as a stopping criterion in our large-scale Monte-Carlo analysis. This might not be necessary in real-world applications to a small number of model instances where also more restarts could be used. Alternatively, one might experiment with so-called global solvers, an option we discarded after tests did not look promising. We conclude from our tests that the heuristics work quite well with pure linear models, but that no general conclusion can be drawn in presence of integers. The model layout with investments decisions for multiple machines in different sizes, required by different sets of crops and depicted as integers, proved quite challenging for larger number of observations, but performed still quite well with one or two.

TROOST and BERGER 2014 discuss an alternative approach using real-world observations to calibrate a larger set of MIP programming models. While they consider somewhat more parameter as uncertain com-

[5] Each trial required from a few to several ten seconds to solve. With 50x20 Monte-Carlos and each Monte-Carlo draw subject to up to 100 restarts with perturbed parameter sets, the time requirements became prohibitive for larger number of observations. One reason was that CONOPT4 went sometimes into infinite loops which required breaking that solve manually such that the processes could not run completely unintended. Generating all results required more than a week despite solving different tests in parallel.

pared to our didactic application example, they consider also 0-1 parameter choices such as "if all agents face either unlimited or zero demand or supply of this good". In total, they tested around 2,400 parameter combinations from which they fixed only those which in all three observation years could improve some aggregate measure of fit across the around 500 farms. For the rest, they conducted large-scale uncertainty analysis simulation experiments. Thus, their approach aims mostly at excluding some parameters from this second step. The advantage of the TROOST and BERGER (2014) approach is firstly that the individual models need not to be technically integrated into one problem as in our approach. Indeed, each single instance of their model has many thousands variables and equations which might exclude a simultaneous bi-level estimation framework. Secondly, their approach can be used with MIP models where coefficients related to integers are considered uncertain. But it clearly cannot calibrate individual models (almost) perfectly.

A detailed comparison to the approach of JAYET et al. (2020) is not possible with the available information. They also consider selected resource coefficients as uncertain – but as it seems not objective function entries - and improve model calibration of each instance independently by testing in some systematic way different parameter sets.

The true data generation process is unknown in real-world application. Data on production processes on farm (crop nutrient requirements, necessary field operations and related machinery and labour needs, feed and stable place requirements etc.), knowledge on likely constraining endowments (land, labour, machinery, buildings, water rights etc.) and assumptions on optimizing behaviour are the elements in building a detailed bio-economic model. This information typically stems from a mix of sources, such as data from official statistics, often at the level of administrative units instead of individual agents, farm accounting data, farm planning handbooks, field experiments, questionnaires, econometric work etc., (cf. the discussion of parameterization in TROOST and BERGER, 2014). Many of the resulting model parameters are not directly observed for the farm instances subject to calibration. Furthermore, observed yields, netput quantities, prices etc. might deviate from expectations used when allocative decisions were taken. That renders it inviting to consider parameters as uncertain, but runs the risk of overfitting model instances during calibration. Still, besides a well-motivated model structure and parameterization, a satisfactory fit against multiple observations provides more or less the only indication to what extent a bio-economic farm-scale model provides a good representation of the real technology and decision behaviour.

Working with algorithms which provide a systematic approach to calibration such as the one proposed in here increases transparency in calibration. Algorithms require clear decisions on what parameters are changed up to what range, and on relative weights used to penalize deviations from given calibration targets and given parameters. Our findings suggest that, firstly, the proposed algorithm is able to recover the true parameters in Monte-Carlo experiments and, secondly, that it enables quite accurate calibration as long as degrees of freedom allow for it. As such, there is some confidence that a parameterization resulting from its application is a reasonable choice against the background of the chosen settings, such as calibration targets and assumed distributions of parameters and errors terms. A further advantage is that it can be entirely encoded in GAMS, a software widely used in field of farm-scale programming models. Indeed, we were able to apply the approach also successfully in the highly detailed bio-economic farm model FARMDYN (BRITZ et al., 2016).

Summary and Conclusion

We present a rather universally applicable algorithm which calibrates a (farm-scale) LP or MIP model drawing on the idea of PMP (HOWITT, 1995) by systematically adjusting uncertain coefficients in the model, but without introducing new structural elements such as non-linear "unobserved" costs. The aim here is to base the allocation steering of the simulation model solely on the interplay of the linear objective and the (rich) constraint set of the model. As such, the proposed approach to calibration is hardly suitable to most models where now PMP is applied. We apply a bi-level estimation framework where the outer problem controls the fit of the calibration and the lower problem is the actual simulation model, represented by its FOCs. Technically, we use the EMP package of GAMS in combination with the NLPEC solver to automatically convert the simulation model into its FOC. Due to the non-convex solution space and the presence of integer variables, the application of a gradient based solver to find local minima is combined with heuristics to generate starting points from existing good solutions.

We exemplify the application with a simple didactic model with a land balance, several labour and maximal crop share constraints; a variant of this model

introduces machinery requirements linked to integers depicting investments in machines. The approach is tested in a systematic way by starting with known model parameterizations and related optimal solutions, for different number of observations. We perturb the optimal solutions by white noise, next apply some data balancing step as one would also expect in a real-world application (e.g. to render observations consistent to maximal crop shares and given land) and next calibrate the model against the resulting observations. For just one observation, we find (almost) perfect calibration in all tests. With growing number of observations, we move away from an underdetermined model such that the algorithm searches a best-fit parameter set, but still, in average, allows for a quite accurate calibration against the perturbed observations. Jointly with the GAMS code of the example found as an annex, the presented approach opens the door for more transparent, systematic and automated calibration of farm-scale programming models. It was also successfully tested with the quite detailed bio-economic farm model FARMDYN (BRITZ et al., 2016) to calibrate it to crop shares and animal herds.

References

ARATA, L. and W. BRITZ (2019): Econometric mathematical programming: an application to the estimation of costs and risk preferences at farm level. In: Agricultural Economics 50 (2): 191-206.

BRITZ, W. and P. WITZKE (2012): CAPRI model documentation 2012. University of Bonn, Institute for Food and Resource Economics, Bonn.

BRITZ, W., M. FERRIS and A. KUHN (2013): Modeling water allocating institutions based on multiple optimization problems with equilibrium constraints. In: Environmental Modelling & Software 46 (Aug. 2013): 196-207.

BRITZ, W., B. LENGERS, T. KUHN and D. SCHÄFER (2016): A highly detailed template model for dynamic optimization of farms-FARMDYN. University of Bonn. Institute for Food and Resource Economics. http://www.ilr.uni-bonn.de/em/rsrch/farmdyn/farmdyn_docu.pdf.

BRITZ, W., M. VAN ITTERSUM, A. OUDE LANSINK and T. HECKELEI (2012): Tools for Integrated Assessment in Agriculture. State of the Art and Challenges. In: Bio-based and Applied Economics 1 (2): 125-150.

BRITZ W. (2021): Comparing penalty functions in balancing and dis-aggregating Social Accounting Matrices. In: Journal of Global Economic Analysis 6 (1): 34-81.

BUYSSE, J., G. VAN HUYLENBROECK and L. LAUWERS (2007): Normative, positive and econometric mathematical programming as tools for incorporation of multifunctionality in agricultural policy modelling. In: Agriculture, ecosystems & environment 120 (1): 70-81.

DE FRAHAN, B.H. (2019): Towards Econometric Mathematical Programming for Policy Analysis. In: Msangi, S. and D. Mac Ewan (eds.): Applied Methods for Agriculture and Natural Resource Management. Springer, Cham: 11-36.

FERRIS, M.C. and GAMS DEVELOPMENT (2009): GAMS/NLPEC Solver Manual. GAMS, Solver Descriptions.

FERRIS, M.C., S.P. DIRKSE, J.-H. JAGLA and A. MEERAUS (2009): An extended mathematical programming framework. In: Computers and Chemical Engineering 33 (12): 1973-1982.

GOCHT, A. and W. BRITZ (2011): EU-wide farm type supply models in CAPRI - How to consistently disaggregate sector models into farm type models. In: Journal of Policy Modeling 33 (1): 146-167.

HECKELEI, T., W. BRITZ and Y. ZHANG (2012): Positive mathematical programming approaches–recent developments in literature and applied modelling. In: Bio-based and Applied Economics 1 (1): 109-124.

HECKELEI, T., R.C. MITTELHAMMER and T. JANSSON. (2008): A Bayesian alternative to generalized cross entropy solutions for underdetermined econometric models. Discussion Paper 2008: 2. Institute for Food and Resource Economics, University Bonn.

HOWITT, R.E., J. MEDELLÍN-AZUARA, D. DUNCAN MACEWAN and J.R. LUND (2012): Calibrating disaggregate economic models of agricultural production and water management. In: Environmental Modelling & Software 38 (Dec. 2012): 244-258.

HOWITT, R.E. (1995): Positive mathematical programming. In: American Journal of Agricultural economics 77 (2): 329-342.

JANSSEN, S. and M.K. VAN ITTERSUM (2007): Assessing farm innovations and responses to policies: a review of bio-economic farm models. In: Agricultural systems 94 (3): 622-636.

JAYET, P.A., A. PETSAKOS, R. CHAKIR, A. LUNGARSKA, S. DE CARA, E. PETEL, P. HUMBLOT, C. GODARD, D. LECLÈRE, P. CANTELAUBE and C. BOURGEOIS (2020): The European agro-economic model AROPAj. UMR Economie Publique INRA-210-AgroParisTech Université Paris-Saclay. https://www6.versailles-grignon.inrae.fr/economie_publique/content/download/3630/38558/version/1/file/ArticlAROPAj.pdf.

JANSSON, T. and T. HECKELEI (2011): Estimating a primal model of regional crop supply in the European Union. In: Journal of Agricultural Economics 62 (1): 137-152.

KANELLOPOULOS, A., P. BERENTSEN, T. HECKELEI, M. VAN ITTERSUM and A.O. LANSINK (2010): Assessing the forecasting performance of a generic bio-economic farm model calibrated with two different PMP variants. In: Journal of Agricultural Economics 61 (2): 274-294.

KUHN, A., W. BRITZ, D.K. WILLY and P. VAN OEL (2016): Simulating the viability of water institutions under volatile rainfall conditions–The case of the Lake Naivasha Basin. In: Environmental modelling & software 75 (Jan. 2016): 373-387.

LOUHICHI, K., P. CIAIAN, M. ESPINOSA, L. COLEN, A. PERNI and S. GOMEZ Y PALOMA (2015): An EU-wide individual farm model for common agricultural policy analysis (IFM-CAP): JRC Report. http://publications.jrc.ec.eu

ropa.eu/repository/bitstream/JRC92574/jrcreport_jrc92574.pdf.

LOUHICHI, K., S. JANSSEN, A. KANELLOPOULOS, H. LI, N. BORKOWSKI, G. FLICHMAN, H. HENGSDIJK, P. ZANDER, M.B. FONSECA, G. STOKSTAD and I.N. ATHANASIADIS (2010): A generic farming system simulator. In: Brouwer, Floor M. and M.K. van Ittersum (eds.): Environmental and Agricultural Modelling. Springer, Dordrecht: 109-132.

MÉREL, P. and S. BUCARAM (2010): Exact calibration of programming models of agricultural supply against exogenous supply elasticities. In: European Review of Agricultural Economics 37 (3): 395-418.

RODRIGUES, J. (2014): A Bayesian approach to the balancing of statistical economic data. In: Entropy 16 (3): 1243-1271.

SCHÄFER, D. and W. BRITZ (2017): Estimating impacts of the revised German fertilizer ordinance on manure transport flows within North Rhine-Westphalia. Poster presented at the EAAE XVth Congress, Parma (Italy).

SCHMID, E. and S. SINABELL (2005): Using the positive mathematical programming method to calibrate linear programming models. Universität für Bodenkultur, Department für Wirtschafts-u. Sozialwissenschaften, Institut für Nachhaltige Wirtschaftsentwicklung, Wien.

TROOST, C. and T. BERGER (2014): Dealing with uncertainty in agent-based simulation: farm-level modeling of adaptation to climate change in Southwest Germany. In: American Journal of Agricultural Economics 97 (3): 833-854.

VICENTE, L.N. and P.H. CALAMAI (1994): Bilevel and multilevel programming: A bibliography review. In: Journal of Global optimization 5 (3): 291-306.

DR. WOLFGANG BRITZ
University Bonn
Institute for Food and Resource Economics
Nussallee 21, 53229 Bonn, Germany
e-mail: wolfgang.britz@ilr.uni-bonn.de

Appendices

Appendix A: Details with Regard to Machinery Requirements and Investments

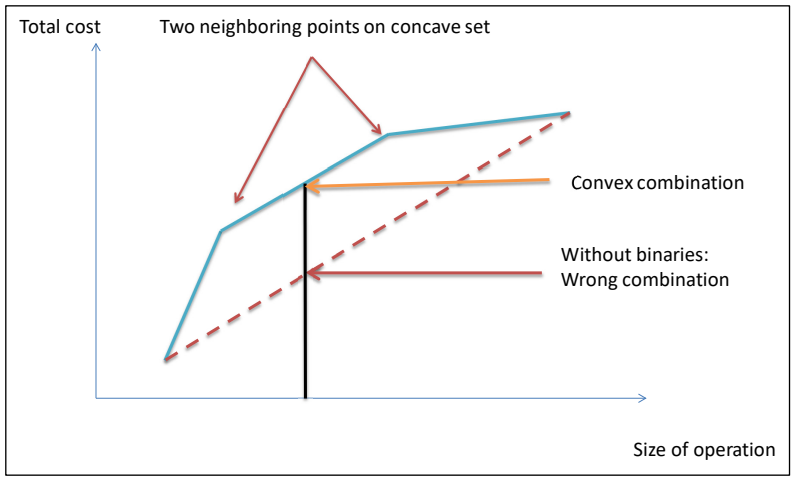
We model increasing return-to-scale in investments into five different machine types t by interpolating linearly based on convex combinations between neighboring points of a concave set, depicting machines of increasing sizes s (see Figure A1) These two neighboring points are endogenously determined by integer variables. Let *machI* depict these integer variables for machine of type t and size s. Let $s'(s)$ denote the set of points which are not neighbors of s. The following equation (1) ensure that (a) if s is selected, only neighboring points can be chosen and (b) if it is not selected, not more than two non-neighbors are active:

$$\sum_{s'(s)} machI_{s'}^{t} \leq 2(1 - machI_{s}^{t}) \qquad (1)$$

The firm can realize based on the fractional variables *mach* a linear combination between the active two neighboring points on the concave set:

$$\begin{aligned} &\mathrm{mach}_s^t \leq \mathrm{machI}_s^t \\ &\sum_s \mathrm{mach}_s^t = 1 \end{aligned} \qquad (2)$$

Figure A1. Convex combination over concave set



Source: authors

The machinery requirements, here expressed in hectares, are covered by the chosen machine mix where related entries in resource constraint matrix A are unity if the crop c requires that type of machine t and zero otherwise:

$$\sum_s \mathrm{mach}_s^t \geq \sum_c x_{c,t}\, A_{t,c} \qquad (3)$$

In case of multiple years, the average annual requirements are defined by the RHS of (3) and the costs per machine multiplied with the number of years in the objective function.

Appendix B: Some Notes on the GAMS Code

The code was tested with the GAMS version 34.2. The necessary option files for NLPEC and CONOPT4 and the GAMS code are available as a zip container. Note that the maximal number of processors used for CONOPT4 is set to 3 (threads = 3) in the option file, a setting which might need adjustment on other machines. As the program generates larger intermediate files, it is recommended to place the GAMS code on a local disk and not in a network. The seed of the random generator in GAMS is fixed which should guarantee that the same sequence of random numbers is drawn when the program is restarted.

The first block of the GAMS code comprises three globals which can be used to generate model variants as indicated by comments, i.e. different number of observations, different variance for the white noise error term used in data generation and, finally, adding or not the integer variables and related equations. The option with multiple farms was not tested in here.

The code produces two GDX files which are dynamically updated during the Monte-Carlo draws. Best*.gdx reports statistics about the individual restarts and comprises the current best variables found in the bi-level estimator. Results*.gdx only reports the final best results from each draw along with statistics summarizes these draws. The file names report the settings used, best_1_5_no.gdx reports for the case of 1 observation, a variance of 5 and no integers. It should be relative straightforward to test the framework for more than the current five crops. Beside extending the set *c*, also the cross-set with the machinery types *machType_crop* would need to be expanded. The code proposes to use CPLEXD to solve the LP or MIP problems. If no integers are used, it can be replaced by a multi-purpose NLP solver such as CONOPT4.

Appendix C: GAMS Code of Didactic Model

See bilevel_capmip.gms and the different options file in the annex, which need to be copied in the same directory. To ease installation, all files are also available as a zip-archive (Code_plus_options_files.zip).