Environmental Modelling and Software 158 (2022) 105556

Contents lists available at ScienceDirect

Environmental Modelling and Software

journal homepage: www.elsevier.com/locate/envsoft

Conditional interval reduction method: A possible new direction for the optimization of process based models

R. Hollós [a,b,c], N. Fodor [b,d,*], K. Merganičová [d,e], D. Hidy [f], T. Árendás [b], T. Grünwald [g], Z. Barcza [a,d]

[a] Department of Meteorology, Institute of Geography and Earth Sciences, ELTE Eötvös Loránd University, H-1117, Budapest, Pázmány P. s. 1/A, Hungary
[b] Agricultural Institute, Centre for Agricultural Research, H-2462, Martonvásár, Brunszvik u. 2, Hungary
[c] Doctoral School of Environmental Sciences, ELTE Eötvös Loránd University, H-1117, Budapest, Pázmány P. s. 1/A, Hungary
[d] Faculty of Forestry and Wood Sciences, Czech University of Life Sciences Prague, 165 21, Prague 6, Kamýcká 129, Czech Republic
[e] Department of Biodiversity of Ecosystems and Landscape, Institute of Landscape Ecology, Slovak Academy of Sciences, SK 949 01, Nitra, Akademická 2, Slovakia
[f] MTA-MATE Agroecology Research Group, Department of Plant Physiology and Plant Ecology, Hungarian University for Agriculture and Life Sciences, H-2100, Gödöllő, Páter Károly u.1, Hungary
[g] Institute of Hydrology and Meteorology, Technische Universität Dresden, D-01737, Tharandt, Pienner Str. 23, Germany

ARTICLE INFO

Keywords:
Model optimization
Bayesian calibration
Parameter constraints
Decision tree

ABSTRACT

Application of process-based models at different spatial scales requires their proper parameterization. This task is typically executed using trial-and-error parameter adjustment or a probabilistic method. Practical application of the probabilistic methods is hampered by methodological complexity and lack of interpretability. Here we present a novel approach for the parameterization of process-based models that we call as conditional interval refinement method (CIRM). The method can be best described as the combination of a probabilistic approach and the advantages of the expert-based parameter adjustment. CIRM was demonstrated by optimizing the Biome-BGCMuSo biogeochemical model using maize yield observations. The proposed approach uses the General Likelihood Uncertainty Estimation (GLUE) method with additional expert knowledge, supplemented by the construction and interpretation of decision trees. It was demonstrated that the iterative, fully automatic method successfully constrained the parameter intervals meanwhile our confidence on the parameters increased. The algorithm can easily be implemented with other process-based models.

1. Introduction

Process based models are typically associated with many parameters that have to be set by the user in any modeling exercise (Therond et al., 2011; Wöhling et al., 2013; Hararuk et al., 2014; Bilionis et al., 2015; Hidy et al., 2022). Model parameterization is a complex procedure as generally some of the adjustable constants of the model cannot be measured directly (e.g. they are empirical coefficients associated with some process representation). Another problem is the well-recognized uncertainty of the model parameters especially if the model is used at large spatial scales (Van Oijen et al., 2005; Xiong et al., 2008; Angulo et al., 2013). Inverse modeling (also referred as parameter estimation or model optimization/calibration) techniques are widely used to estimate unknown or uncertain model parameters based on observations (Braswell et al., 2005; Van Oijen et al., 2005; Sadegh and Vrugt 2013; Bilionis et al., 2015).

Inverse modelling is considered successful if and only if the model results based on the estimated parameters are reasonably close to the observations that represent the "reality" (with some noise quantified by the measurement error). If the uncertainty of the estimation is also low, the model is considered appropriate to describe the studied system accurately and precisely. The key factor here is the metric for the "closeness" of the simulations to the observations, and the uncertainty of the parameter estimation (where the latter can be represented by a probability distribution; Trudinger et al., 2007). The metric is represented by the objective function (e.g. likelihood for probabilistic methods), while the latter is quantified by the uncertainty range (i.e. posterior density function for probabilistic case). Up to recently, most of the inverse modelling efforts directed toward only the objective function, without estimating the parameter uncertainties (e.g. Wallach et al.,

[*] Corresponding author. Agricultural Institute, Centre for Agricultural Research, H-2462, Martonvásár, Brunszvik u. 2, Hungary.
E-mail address: fodor.nandor@atk.hu (N. Fodor).

https://doi.org/10.1016/j.envsoft.2022.105556
Received 2 May 2022; Received in revised form 16 August 2022; Accepted 10 October 2022
Available online 13 October 2022

2021). Derivate-based optimization techniques such as the Levenberg-Marquardt method are frequently used (Moré, 1978; Trudinger et al., 2007; Brunetti et al., 2022). These techniques can provide fast convergence with low stability (ideal for deep learning applications, where these are most commonly used). The success of the procedure depends on the initial values of the parameters and the choice of meta parameters (e.g. the learning rate for gradient descent; Goodfellow et al., 2017, p. 101; Tarantola, 2005, p. 76). For complex process-based models information about the derivate is usually not available, therefore the so-called derivate-free methods are preferred.

Two subcategories can be distinguished within the derivate-free methods: the direct search-based methods and the probabilistic methods. The first one is only suitable for finding the optimum parameter values and more subject to the "curse of dimensionality" (Bellman, 1957) meaning that these methods are applicable only in low-dimensional cases (Tarantola, 2005, p. 42). When we only have a few parameters to optimize, grid search can be fast, simple and stable, meaning that after repeating the optimization with the same hyperparameters (i.e. number of dividing points), the result would be close to the original optimization output. Metaheuristic algorithms such as genetic algorithms can be used for addressing the "curse of dimensionality", but usually by sacrificing the stability or the unicity of the result (Gogna and Tayal, 2013; White et al., 2022). Finding the global optimum is also not guaranteed in such case.

The goal of the probabilistic methods (which is the subject of this study) is usually not only to find the optimum of the parameters, but also to provide some uncertainty for these (Van Oijen et al., 2005; Hartig et al., 2011). These methods are largely flexible and naturally provide possibilities to calibrate input parameters while considering modelling, parametrization and measurement errors at the same time (Tarantola, 2005). However, flexibility has its cost. The more flexible the system is the more hyperparameters and assumptions have to be used in order to simplify the workflow. Building up a probabilistic model is generally much harder than finding a good objective function.

The least complicated probabilistic method that can be used for estimating the models' input parameters is the maximum likelihood (ML) estimation. This method relies on an efficient sampling algorithm, which effectively samples from the input parameter space. In this context effectivity means that the sampling distribution resembles the prior distribution with the minimum number of sampled parameters. Depending on the effectiveness of the sampling procedure, the "curse of dimensionality" can be a major problem here as well. However, the biggest problem is that if the number of sampled parameters is low, the confidence over the ML value is unreasonably strong, and usually the ML values fail on independent data. There are two possible directions to address this issue: the frequentist and the Bayesian approach. The frequentist solution is regularization. For example, in the General Likelihood Uncertainty Estimation (GLUE) method, not all likelihood values are used for constructing the distribution of the estimated input parameter values, but only the so-called behavioral values, which are usually defined as the top x % (typically x = 5) of the sampled parameters (Prihodko et al., 2008; Beven and Binley, 2014; Sexton et al., 2016). The behavioral parameter values are all considered as "good" parameter values. Among them the parameter value with the ML value is just one of the many. The median of the sampled parameter values is frequently considered as the optimum parameter value with the uncertainty interval defined by the empirical cumulative distribution function (ECDF). If the empirical probability density function is unimodal, this approach is more stable and usually gives better results during validation with independent data than the ML estimation.

The other large family of the probabilistic methods is the Bayesian one. These methods are generally more flexible than the frequentist approaches and naturally provide interfaces to (re)use previous knowledge about the parameter space. The use of previous knowledge prevents the optimization algorithms to be overly confident about the estimation which is based on small amount of data points. The Bayes rule is the unifying device with which the prior knowledge can be updated by the likelihood function to gather posterior knowledge (Gelman et al., 1995). The posterior density function provides us with probabilistic information about our parameters. For example, one can answer questions like what is the probability of a given parameter value present within a certain range? With this information we can estimate a parameter with its characteristic points such as a maximum posterior value (MAP) while providing information about the uncertainty intervals like highest posterior density interval (HPDI). However, if our only expectation from the method is to produce reliable prediction, instead of using the MAP value, Bayesian Model Averaging (BMA) would be the best solution (Hinne et al., 2020), as it uses all information from the posterior distribution to predict the output compared to the simpler ensemble method. Furthermore, the posterior distribution is directly useable for hypothesis testing for example through Bayesian Factors (BF) (Gelman et al., 1995).

Despite all of their strengths – if executed properly – Bayesian methods are undoubtedly complicated. Thus, during the inverse modelling process excessive number of assumptions and numerous choices have to be made. First, proper likelihood and prior functions have to be chosen. The choice is always connected to the problem, but not always obvious. For example, universally applicable uninformative prior distribution does not exist (Pericchi and Walley, 1991), although many use a uniform prior distribution for this purpose due to its simplicity (Pericchi and Walley, 1991; Gelman, 1996; Gelman and Yao, 2020; Wallach et al., 2021). Furthermore, in spite of the fact that choosing the proper likelihood function is one of the most important decision to be made (Trudinger et al., 2007; Dumont et al., 2014), modelers still usually use normal likelihood (Wallach et al., 2021). However, choosing the likelihood function family is still not sufficient to define the right priors or the global likelihood function. There are several additional assumptions, such as the independency of measurements, and also the independency of the measurement and model errors (Tarantola, 2005). The most frequent assumption by far is that gaining reliable inference about the input parameter space is possible by examining only the marginal posterior distributions. There are few attempts to deal with the interdependence of the posterior distribution, but these are usually examined pairwise or in a few cases by 3D scatter plots (Sadegh and Vrugt, 2013; Beven and Binley, 2014; Her and Chaubey, 2015) which clearly cannot capture higher dimensional relationships. The assumptions about the sampling procedure are similar: most of the proposed values for the parameters are sampled independently.

Depending on the dimensionality of the parameter space probabilistic calibration is computationally expensive. Thus, the methods which sample less while getting the most informative results are preferable. If input parameter dependence is taken into account, less simulations are needed (Bloom and Williams, 2015). For example, if the model has two input parameters and the sum of the parameters must equal 1, the probability of getting the affordable parameter with random uniform sampling is virtually 0. Because convex relationships are the most widespread, using a sampling method designed specifically for convex cases can have a huge positive impact on the effectivity of the procedure. Otherwise, every concave region can be split into disjoint convex regions, where the input parameters can be sampled independently. If the convex region is not a polytope, it can be approximated with polytopes. Therefore, techniques used to sample convex polytope regions are general enough to be good candidates to sample any parameter region. One of these algorithms is the Hit and Run sampling (Lovász and Vempala, 2003), which (according to our best knowledge) has not been used for process based models yet.

At this point it is important to note that it is still an open question whether the Bayesian method has advantages over the frequentist method even if Bayesian methods are not used properly (Gelman and Yao, 2020). Many studies focus on the practical application of different methods but we are very far from offering "best practices" or universally applicable solutions to the modeler community.

There are attempts to improve the success of optimization. Because of the complexity of the models and the related high dimensionality of the parameterization, more and more observations are needed to confidently perform inverse modeling, and most of the time the problem is still underdetermined (Sadegh and Vrugt, 2013). The modeler can theoretically tackle this problem by introducing additional information about the modelled system. Providing information about the relationship of the input parameters is a recently proposed new direction (Richardson et al., 2010; Bloom and Williams, 2015). This approach is based on the recognition that not every parameter combination is feasible based on the expert knowledge.

Conditioning the input parameter space is not the only option to improve the results of model optimization. Process-based models are inherently complex and produce more than one output streams. Calibrating models considering only one of these can result in "good results for wrong reasons" (Beven and Binley, 2014). As a consequence, with the resulting parameterization the model can predict the given variable reasonably well on the training data but might fail on the validation data (that is the typical case with overfitting). The chance of overfitting is the highest if the parameters are associated with equifinality which means that in the predefined interval the likelihood of different parameter values is similar. In the case of equifinality we cannot choose objectively from the posterior parameters which means that the parameter uncertainty cannot be constrained (Beven and Freer, 2001). Equifinality does not necessarily mean failure of the simulation but rather it is a limitation of the inversion method.

One possible solution to avoid equifinality is the application of multi-objective calibration. By using multiple objective functions based on multiple observation data streams the result can be more stable with fewer parameters in equifinality (Her and Seong, 2018). However, multi-objective calibration needs good quality observation data streams to ensure a successful inversion. Because of the output stream interdependence, adding one more data stream to the process can lead to a complicated workflow. The noise of multiple data streams is not additive (c.f. error covariance matrix). In real life cases generally we do not have enough data for multi-objective calibration, or the data is too noisy.

In conjunction with multi-objective calibration, the modeler can use some rejection filtering techniques. One of the main advantage of these approaches is the resilience against correlations between the different statistics of the model outputs (Hartig et al., 2011). However, applying rejection filtering alone does not shrink the parameter space, and these methods cannot be easily interpreted, hence the modelers who traditionally use trial-and-error approach with (implicit) rejection filtering do not have enough control over the inverse modeling procedure. Furthermore, if the ratio of the filtered, acceptable simulations is too low, the convergence of the posterior sampling methods can be exceptionally slow or even questionable.

The complexity of the implementation of probabilistic methods and the possibility to get unusable results (i.e. equifinality) might be the main reasons why the majority of the researchers still prefer the trial-and-error method in model optimization (Wallach et al., 2021). Clearly, there is a great need to develop interpretable, easy-to-use and effective methods that support modelers to improve the optimization of the models especially in high dimensional cases.

In this study we introduce a novel method which can resolve some ongoing issues that are described above in probabilistic, process-based model optimization. The method uses observations supplemented by expert knowledge on the simulated system. Unlike in case of the previously proposed methods (Richardson et al., 2010; Bloom and Williams, 2015) conditioning (rejection filtering) on the output data streams is introduced in our novel method. A decision tree based algorithm is presented that reduces a priori input parameter intervals during the calibration of deterministic models and increases the reliability of simulation results of the calibrated model. In this study we apply the GLUE method for its simplicity, supported by an implementation of the Hit and Run algorithm. Our method can be easily combined with any Bayesian or frequentist inverse modeling method and can contribute to improved and more efficient model optimization results in an array of scientific disciplines.

2. Material and methods

2.1. Experimental data

In the present study maize yield observations were used from Martonvásár (47°19′56.97"N, 18°47′50.61"E), Hungary (Central Europe). The climate of Martonvásár is continental with Mediterranean and oceanic influences. Mean annual temperature is 11.2 °C, while long term mean precipitation is around 550 mm. According to the FAO-WRB classification system (IUSS Working Group, 2015), the soil type of the area is a Haplic Chernozem, with an average of 51.4% sand, 34% silt and 14.6% clay content. Bulk density is 1.47 g cm^{-3}, pH is 7.3, $CaCO_3$ content is 0–1%, and the mean soil organic matter content in the topsoil is 3.2%. The plant-available macronutrient supply in the soil is poor for Phosphorus and medium to good for Potassium, based on the ProPlanta plant nutrition advisory system (Fodor et al., 2011). In the LTFEs every treatment is arranged in a random block design with 20–40 m^2 plots in four replicates.

Experimental data were collected in long-term field experiments (LTFE) that were set up at the Centre for Agricultural Research in the 1960s. Maize is grown in several LTFE sites within and around the city of Martonvásár. Some of the experiments focus on the effect of organic and mineral fertilizer application on crop yield. Others focus on the effect of soil cultivation, hybrid selection, planting date and plant density on crop physiology and yield. In the present study, a composite maize yield dataset was used by retrieving the yield data of FAO350-400 hybrids in high Nitrogen level (at least 160 kgN ha^{-1} year^{-1}) treatments of four LTFEs, characteristic to the Hungarian maize production system in the 1994–2018 period (plant density: 70.000 plant per hectare; planting date: second decade of April; harvest date: mid-October). The average yield was calculated from 16 yield data of the four selected LTFEs (with 4 repetitions in each treatment with the parcel size of 20–40 m^2) for each year. Average maize yield (calculated from all plots and all years) in the experiments was 7.7 t ha^{-1} with a relatively large uncertainty (SD = 1.58 t ha^{-1}; min = 0.96 t ha^{-1}; max = 13.57 t ha^{-1}). Note that the observations refer to dry matter of the grain yield.

NUTS3 level (EuroStat, 2021) maize yield data from Fejér county (where the experimental site is located) was also used in the study. Census data were retrieved from the database of the Hungarian Central Statistical Office for the period of 1991–2018. The average maize yield for the study period was 6.36 t ha^{-1} in Fejér county (SD = 1.92 t ha^{-1}; min = 2.89 t ha^{-1}; max = 10.06 t ha^{-1}).

2.2. Biome-BGCMuSo biogeochemical model

The novel method introduced in the present study was linked with the Biome-BGCMuSo process-based model. Biome-BGCMuSo is a general purpose, process-based, biogeochemical model that simulates the full carbon, nitrogen and water budget of terrestrial ecosystems (Hidy et al., 2012, 2016, 2021, 2022). Biome-BGCMuSo is a branch of the well-known Biome-BGC model (Running and Hunt, 1993; Thornton, 1998; Thornton et al., 2002; Churkina et al., 2009; Di Vittorio et al., 2010). Biome-BGC was significantly improved and extended in many terms relative to the original model. Developments addressed soil processes, introduction of management options, quantification of disturbance effect on plant physiology, and many other processes. One major milestone of the model development was the construction of a 10-layer soil submodule with sophisticated soil water balance routine, the layer-by-layer representation of C and N dynamics within the soil, and the implementation of detailed nitrification/denitrification routine. Improvement of different stress factors (drought stress, nitrogen stress, heat stress) was also a major improvement. Growing-degree-day based, phenophase specific allocation option was also included that enabled the detailed simulations of crops. In cropland simulations detailed management information is essential for proper results (including the timing and amount of applied fertilizer, planting date, soil cultivation type and date, harvest date, residue management). The present Biome-BGCMuSo model can be considered as a combined biogeochemical-crop model with state-of-the-art process representation at both scientific fields.

In this study Biome-BGCMuSo v6.3 was used. Detailed description about the developments can be found in Hidy et al. (2012, 2016, 2022), Fodor et al. (2021), while additional details are available in the User's Guide (Hidy et al., 2021).

The parameterization of Biome-BGCMuSo is a complex task not only because of the high number of adjustable ecophysiological parameters, but also because of the existence of some rules (i.e. constraints) that have to be fulfilled in order to ensure a successful simulation (Hidy et al., 2021). For some parameters these rules are relatively simple (e.g. C:N ratio of leaf litter must be greater than the C:N ratio of leaves), but in some cases they are relatively complex (e.g. the sum of the parameters that control the allocation of carbon to the different plant compartments must sum up to 1). These rules complicate the optimization of the model since in the Monte Carlo framework random numbers are generated within predefined intervals of selected parameters that drive the model. Clearly, for example in the case of the allocation parameters random numbers will not sum up to 1 in the majority of the cases which will result in a large number of unsuccessful simulations.

2.3. Parameterization and model setup

2.3.1. A priori parameterization

For the construction of the a priori maize ecophysiological parameterization literature search was conducted first for maize related data such as specific leaf area, maximum stomatal conductance, canopy light interception coefficient etc. In the case of the empirical parameters (Hidy et al., 2021) optimization was performed using a multi-step procedure. Parameter adjustment was performed based on eddy covariance data (gross primary production (GPP) and evapotranspiration (ET)) from the Klingenberg cropland site (DE-Kli FluxNet code, 50°53′35"N; 13°31′20.6"E) in Germany (Prescher et al., 2010) based on the maize years (2007, 2012 and 2018). Additionally, leaf area index (LAI) data was also used from the site for years when it was available. Phenological phase dependent allocation parameters were set based on a manual (trial and error) adjustment. Parameters for the Penman-Monteith equation based evapotranspiration routine were also set using eddy covariance ET data from Klingenberg.

As part of previous modeling exercises, maize parameterization was further evaluated previously and adjusted at other data rich experimental sites in the USA (Bushland lysimeter site in Texas, and Mead eddy covariance site in Nebraska (US-Ne2 and US-Ne3 FluxNet codes)). It means that maize parameterization was tested for sites with different climatic conditions, management types and with different maize cultivars characterized by a diversity of FAO numbers. The model optimization/validation efforts revealed that it is not possible to construct a single, universally applicable maize parameterization that is useable in different climatic and agro-management conditions. Some of the parameters were highly site (and in some cases year) dependent (senescence related parameter, allocation parameters and plant tissue lifetime related parameters). These previous experiences paved the way for the a priori parameterization of the model.

Unfortunately, at present there is no maize related eddy covariance data available for Hungary in spite of the fact that 3 sites are running at present above croplands. It means that optimization of the model is highly needed based on other available data.

2.3.2. Model setup

Biome-BGCMuSo was run at a plot level simulating a single generic maize parcel at Martonvásár using 220 kgN ha^{-1} y^{-1} mineral fertilizer amount applied at the beginning of April. Planting and harvest dates were set according the reported dates. Driving meteorological data was retrieved from the FORESEE database (Open Database FOR ClimatE Change-Related Impact Studies in CEntral Europe) which is a free meteorological database for Central Europe with 0.1° × 0.1° spatial resolution (Dobor et al., 2015).

The soil input file of the model was constructed based on observations about the soil texture and soil water retention curve measurements using the pipette method (ISO 11277) on disturbed and sand/kaolin-box method (ISO 11274) on 100 cm^3 undisturbed core samples collected from the LTFEs. For most of the nitrogen cycle parameters values proposed by Hidy et al. (2021) were used.

For the NUTS3 level simulations the FORESEE database provided the meteorological data. The DOSoReMI database (Pásztor et al., 2020) was the source for the gridded soil data. The simulations for the Fejér county were performed on a predefined 0.1° × 0.1° resolution grid that was used in Fodor et al. (2021). Fertilization was set according to data from the Hungarian Central Statistical Office. Planting data was 15 April, and harvest date was 10 October in all simulations. We used 51 grid cells covering an area of 4358 km^2. The simulated yield data was aggregated at the county level by simple averaging of cell-specific yields per year.

2.4. Description of the conditional interval reduction method

2.4.1. Problem overview

Let $\mathscr{M}:\mathscr{S}\rightarrow\mathscr{O}$ be an arbitrary process-based model that estimates some output data ($\mathscr{O}$) based on some input parameters ($\theta\in S$), where $\mathscr{S}\subset\mathbb{R}^n$ is the n dimensional input parameter space, and $\mathscr{O}\subset\mathbb{R}^{l\times k}$ is the matrix of model output, where l is the number of time steps and k is the number of output data streams (representing the simulated variables). Note that in this study the word 'parameter' means adjustable input data. In some cases, convex constraints have to be defined over the $\mathscr{S}$ (i. e. to handle dependencies between the input parameters; see Section 2.2). This can be done by introducing matrices (G, E), and vectors ($\overline{e}$, $\overline{h}$) for which

$$\begin{aligned} \boldsymbol{G}\theta &\leq \overline{h} \\ \boldsymbol{E}\theta &= \overline{e} \end{aligned} \tag{1}$$

In the present study **G** controls the parameter ranges and the dependencies between parameters with relations (i.e. one parameter must be smaller than the other one; see Section 2.2). **E** in this context is used for the allocation related parameters where the sum of all considered parameters must equal 1 (Hidy et al., 2021), but of course the applicability of the constraint can be more general (Bloom and Williams, 2015). Supplementary material contains the mathematical representation of the above-described, matrix-based parameter dependencies with two examples.

Inverse modeling has two major objectives (Tarantola, 2005). The first one corresponds to the practical application of the model which needs proper parameterization to achieve reliable simulation results that are in optimal agreement with the observations. This aim means that we need point estimation for the parameters (single parameter set for all $\theta\in S$). The second one is the estimation of the uncertainty intervals of the parameters for future applications. The intervals indicate our imperfect knowledge of the parameter values (van Oijen et el., 2005). Uncertainty intervals can be used to define uncertainty ranges of the simulated output variables.

Given the observations ($\mathscr{D}$) about the modelled system with predetermined uncertainties and the scientific knowledge of the environment, the modeler can sample from a probability density function (pdf) for the model input parameters ($p(\theta|d), d \in \mathscr{D}$). During the sampling procedure the model is evaluated against $\mathscr{D}$. This pdf is the so-called posterior function, and intuitively if the parameter space can be described by a continuous random variable, it is the probability that θ lies in the interval of $(\theta, \theta + \varepsilon)$ given the measured data d, where ε is an infinitesimally small vector. Using the posterior distribution we can achieve both objectives defined above.

Usually, the posterior distribution cannot be sampled directly, because its distribution is not known (previous experience might be invalid in new optimization exercises; van Oijen et al., 2005). However, when the likelihood function ($\mathscr{L} = p(d|\theta), d \in \mathscr{D}$; that is a probabilistic model for the model error; Gelman et al., 1995) and the prior knowledge $p(\theta)$ (called prior distribution) is given during a Monte Carlo (MC) experiment, the Bayesian rule can be used for updating:

$$p(\theta|d) = \frac{p(\theta|d)p(\theta)}{p(d)} \tag{2}$$

$$p(d) = \int p(\theta)p(\theta|d)\mathrm{d}\theta = c \in \mathbb{R} \tag{3}$$

Because $p(d)$ is a constant $p(\theta|d) \sim p(\theta|d)p(\theta)$. Usually in the frequently used Markov Chain Monte Carlo (MCMC) sampling, *p(d)* is eliminated by a division (after the detailed balance equation). In a Bayesian workflow it is rather typical that the modeler chooses a uniform prior ($p(\theta_i) \sim U(I_{i,1}, I_{i,2})$), although this choice can be inadequate because uniform prior is not invariant under reparameterization therefore sensitive to the choice of its defining lower ($I_{i,1}$) and upper bounds ($I_{i,2}$).

From the practical point of view, choosing the right prior can help to avoid overfitting by regularizing the likelihood functions. Using uniform prior adds no regularization, thus the maximum posterior values will be the maximum likelihoods, leading to inappropriately strong conclusions (Gelman, 1996). In the case of high amount of observation data, the prior choice is less important, and the maximum likelihood estimates are (approximately) the same as the maximum posterior estimates independently from the choice of the prior function.

Despite the obvious drawbacks, uniform priors are frequently used in Bayesian frameworks, because of their simplicity, and because of the fact that it is easier to think about the parameters in a bounded space (Gelman, 1996; Stedinger et al., 2008; Gelman and Yao, 2020; Wallach et al., 2021). One other possible way to avoid inappropriately strong conclusions is to use other methods of regularization. For example in the Generalized Likelihood Uncertainty Estimation method (GLUE; Prihodko et al., 2008; Beven and Binley, 2014) likelihood filtering is applied. Only the likelihood values above the previously determined quantile (95th percentile is common choice) are retained and considered as behavioral, and the optimum is considered as the median of the filtered parameters. The key advantage of this approach is its simplicity and flexibility, and some degree of resistance against overfitting (note that the method is sensitive to the choice of the "quantile" meta parameter). This method is not considered as a Bayesian one, because we cannot interpret the results as probabilities; it means that we do not have the proper posterior density function, albeit it can be easily extended to be an Approximate Bayesian Computation (ABC) method (Sadegh and Vrugt, 2013).

In this paper a novel method is introduced that can be used for updating prior distributions. Here we used the GLUE method for demonstration purposes and for simplicity (exploiting also the typical visualization method for GLUE in the form of the so-called dotty plots where equifinality can easily be recognized based on the marginal distributions). In the following section a detailed description of the new method is provided.

2.4.2. *Core logic of the output constraint approach*

Let $f : \mathscr{O} \rightarrow \{0, 1\}$ be a function where $f \circ \mathscr{M}$ categorizes model results into two classes: feasible (1) and infeasible (0) output. Feasible in this context means that $\mathscr{O}$ is in accordance with expectations of the modeler. We would like to stress that feasibility is not judged directly based on the quantitative comparison of the observation and the simulation but rather it is based on some additional knowledge about the simulated system. This kind of knowledge can originate from the scientific literature, from almanacs, or from the everyday practice of the modeler. The model has to be set so that $\mathscr{O}$ contains information about the process that is evaluated by f.

For a given θ parameter

$$\theta \in S \text{ is feasible} \Longleftrightarrow (f \circ \mathscr{M})(\theta) = 1 \tag{4}$$

For simplicity f is called the output conditioning (or filtering) function. As we are interested only in parameter values that provide feasible output (hereafter referred as feasible parameters), f is used to filter out infeasible parameter combinations (constraining the sampling procedure from the initial set).

Although the classifier can help filtering out unrealistic simulations, the ratio of the number of "good" simulations compared to all simulations (c_r) can be low in some cases. This ratio is defined as

$$c_r = \frac{\#\, \textit{feasible simulations}}{\#\ \text{all simulations}} \tag{5}$$

This means that the parameter intervals defined by the posterior distribution contain a lot of parameter values that represent "good results for wrong reasons", which means that we cannot use the whole distribution for gaining scientific knowledge because of the lack of confidence in getting feasible and stable results. We need a further inspection of our parameter values and understand why we get a lot of infeasible results (note that from this point parameter values associated with infeasible results are referred as infeasible parameters). In such cases the researcher has to make an effort by manually looking into the parameter intervals and the input parameters to search for possible mistakes on setting the prior intervals or running more simulations to get more feasible output values.

In our approach $f \circ \mathscr{M}$ is considered as a black-box classifier of the input parameter space which is one of the most fundamental recognitions of the presented algorithm. If we approximate $f \circ \mathscr{M}$ with a white-box classifier (*g*), the interpretation of g is isomorphic with the interpretation of $f \circ \mathscr{M}$. Isomorphic in this context means that decisions made by the white-box classifier are the same as those provided by the black-box classifier. Thus, we can use g to modify the prior input parameter intervals to achieve the high c_r.

The simplest yet still flexible and interpretable white-box classifiers are decision trees (DT). This is the reason why they have recently been used for interpreting the decision-making procedure of deep neural networks, and support vector machines (Di Castro and Bertini, 2019; Lee and Kim, 2016). Decision trees used for classification can capture complex relationships between the feature space and the output categories (i.e. discriminate feasible and infeasible parameter values). As a result, they are not much sensitive to collinearities, and they can make complex relationships easily interpretable too.

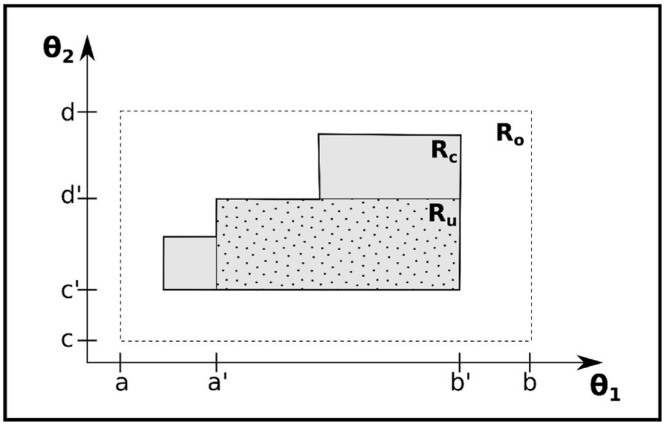

Fig. 1. Overview of the Decision Tree-based updating algorithm on an example of a two-dimensional parameter space. R_o is the original parameter space defined by the parameter intervals [a, b] for θ_1 and [c, d] for θ_2. After applying the constraints, the decision tree defines the constraint region R_c (grey region). The original interval can be updated by selecting the largest rectangular region (R_u) which is inside of R_c. The updated interval that corresponds to R_u for θ_1 is [a', b'], and [c', d'] for θ_2

In our method the so-called CART algorithm (James et al., 2013) was used to generate the trees because of its simplicity. Note that other algorithms such as ID3, C4.5, C5.0 (James et al., 2013) could have been also used. Gini impurity index (Gi) was used to quantify the impurity of a split in making the DT, while the complexity based pruning was used to avoid overfitting (Nobel, 2002). After creating the DT, the accuracy (A) was determined in the following way:

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

where TP means number of true positive cases, FN means number of false negative cases, FP means number of cases identified as false positive while TN means number of true negative decision for the $f \circ \mathscr{M}$.

Low accuracy ($A <$ 0.5) means that the model performance was worse than the random decision. In the case of low performance ($A <$ 0.7) the algorithm is not applicable to parameter-interval change. If the accuracy is high, we can trust our white-box model since it behaves similarly as the black-box.

The leaf nodes of a decision tree are continuous domains in the parameter space defined by its corresponding decision nodes. If $R_1, R_2, \ldots, R_b$ are the input domains for leaf nodes $y_1, y_2, \ldots, y_b$, we are interested in the domains with the highest number of feasible parameters. If there is more than one region with the highest number of feasible parameter values, we can have 3 possibilities to decide which one is the region of interest:

1. Chose the region with the least amount of connected decision nodes and apply the validation to this region
2. Chose the region with the lowest FP rate and apply the validation to this region.
3. Choose both above mentioned regions and apply validation in the next step

With this method, the selected region (R_u) is defined by the decisions at the decision nodes. The region is the maximum-volume-hyperrectangle inscribed into the constrained region (R_c; sides parallel to the parameter vectors) over the original parameter region (R_o) (Fig. 1). The updated parameter intervals are the orthogonal segments which define the hyperrectangle.

If we have multiple output-constraint functions, we create multiple DTs and go through the updating process sequentially. If there is an inconclusive update, we roll back the changes and continue the procedure. In the next step we can shrink the intervals further if there is more information available for that region. The intervals specified by applying individual constraints must overlap to provide updated parameter intervals for the next step.

Note that since in this paper the issue with the similar sized feasible region was not addressed, the above mentioned solution is not part of the current implementation.

2.4.3. Model optimization

Here we combine the GLUE frequentist model optimization method (Prihodko et al., 2008; Stedinger et al., 2008; Beven and Binley, 2014) with the application of the above-described, DT-based classification method. We named this procedure as conditional interval refinement method (CIRM).

It is easy to extend the $f \circ \mathscr{M}$ classifier to support multiple output-constraint functions. If we have m output-constraint functions ($f_i, i \in 1, \ldots, m$) we can define f as the following:

$$f(x) := \prod_{i=1}^{m} f_i(x) \tag{7}$$

After the parameter sampling and model simulations, behavioral parameter choice can be done with quantile filtering (described above) with an additional filtering according to the conditioning (retaining feasible parameter values where $f(x) = 1$).

It is assumed that the modelization uncertainties are negligible here compared to observational uncertainties, and observational uncertainties follow a normal distribution. Furthermore, if the observations are independent from each other the likelihood function is defined as:

$$\mathscr{L}(\theta | d \in D) = \prod_{i=1}^{n_d} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\mathscr{M}(\theta) - d_i}{\sigma}\right)^2} \tag{8}$$

where σ refers to observation uncertainty. For practical considerations (e.g. arithmetic underflow) we used the loglikelihood instead of the likelihood (note that in this case the maxima places are unchanged). For the prior function uniform prior on convex polytope was used. In order to support successful Monte Carlo experiment a novel algorithm was developed and applied here based on the Hit and Run algorithm with mirroring optimization (Lovász and Vempala, 2003; Meersche et al., 2009).

Note that model optimization with GLUE (or with any other optimization method) can be applied at each step of the proposed approach. It is mandatory to perform it only for the last step.

2.4.4. Summary of the proposed method

Algorithm 1 presents the proposed constrained calibration method. The workflow summarizes the methods detailed above also revealing the consecutive steps with all related input and output data. In this paper, for demonstration purposes we performed GLUE in each iteration step.

Algorithm 1. (*Workflow of the proposed method including model optimization and parameter interval update. Meaning of the symbols is defined in the text*)

Full algorithm

```
Input    : $[I_{.,1}, I_{.,2}]$ , $\mathcal{D}, \mathcal{M}, \mathcal{S}$, $\mathbf{E}, \mathbf{G}, \overline{e}, \overline{h}$, $\mathcal{L}, f, N_s$, k = 1, $N_i$
Step 1.  : Sampling $\theta_i$ and evaluate $f \circ$ M, calculate likelihoods for each simulation
Step 1a. : Perform Hit and Run sampling to get $N_s$ feasible $\theta_i \in \mathcal{S}, i \in \{1, \ldots, N_s\}$ for which
               $\mathbf{E\theta} = \overline{e}$
               $\mathbf{G\theta} \leq \overline{h}$
Step 1b. : Run the model simulations and calculate the likelihood values
               $\sigma_i = \mathcal{M}(\theta_i)$
               $l_i = \mathcal{L}(\sigma_i, \mathcal{D})$
Step 1c. : Evaluate $f$:
               $y_i = f(\sigma_i)$
Output   : Sampled parameters and their $y_i$ categories (0 or 1), and the corresponding
           likelihood values
Step 2.  : Calculate $c_r$using eq. (5)
Step 3.  : Train Decision Trees for all output constraint functions and all $y_i$
Step 4.  : Update parameter intervals based on the Decision Trees sequentially
Input    : Current parameter intervals, Decision Tree
Step 4a. : Search for the leaf node with the highest number of successful cases from the
           training set. If there are more than one candidate, choose the leaf node defined by
           the lowest number of decision nodes and prefer the leaf node with the lowest False
           Positive Rate (FPR)
Step 4b. : Modify the parameter ranges according to the decisions which defines the node
           selected in the previous step, if possible
Output   : Updated parameter intervals
Step 5.  : if $k = N_i$, go to step 6. Calculate$c_r$; if $c_r$ is high enough jump to step 6, otherwise
           jump to step 1. with the modified interval and increase counter $k$ by 1
Step 6.    Perform GLUE or other optimization method using the modified parameter ranges
Output   : Updated parameter intervals, GLUE optimum, $c_r$
```

2.5. Implementation of the method

We used the above proposed method to optimize Biome-BGCMuSo v6.3 for maize in a low-data situation (which means low amount of good quality observation data) when only final crop yield data was available, supplemented with some additional information about overall crop properties (that represent constraints.

Biome-BGCMuSo has 120 maize-related ecophysiological parameters that have to be set by the modeler prior to the simulations. Some of the parameters are generic plant parameters (like C:N ratio of plant compartments; maximum stomatal conductance; maximum rooting depth; root distribution parameter; canopy water interception coefficient etc.), while some of them are specific to maize (parameters affecting e.g. heat stress during anthesis, germination as the function of soil water content etc.; Hidy et al., 2021). Among the 120 parameters 42 are affected by some rules (see above). 28 out of the 42 rule-affected

Table 1
Complete list of model parameters that were selected for optimization with the prior parameter ranges. MIN represents $I_{.,1}$, MAX represents $I_{.,2}$

Abbreviation	Description	MIN	MAX
Rubisco	The percent of leaf N content in the Rubisco enzyme. It controls potential rates of carboxylation (White et al., 2000)	0.07	0.12
root_distribution_parameter	Empirical parameter to calculate the distribution of roots within the soil layers (Jarvis, 1989)	2	6
root_weight_to_max_root_depth	This parameter is used in the empirical rooting depth calculation of plants. The parameter controls the dependence of rooting depth on the allocated carbon pool associated with the root system.	0.08	0.14
root_depth_function_shape	This parameter is used in the empirical rooting depth calculation of plants (based on the method of 4M model). The parameter controls the temporal profile of the root depth (can be convex or concave as the function of time).	0.4	1.6
senescence_coeff_for_leaf	Soil moisture stress related mortality coefficient that controls the extent of leaf senescence (fraction of leaf tissue that dies during one day due to stress caused by a prolonged drought event).	0.001	0.04
water stress_effect_on photosynthesis	Empirical parameter that controls the non-stomatal soil water content (SWC) stress effect on photosynthesis (i.e. down-regulation of carbon assimilation).	0	0.7
length_of_phenophase_3	Length of the 3rd phenophase expressed in growing-degree-days (GDD). In the case of maize this refers to early vegetative growth.	240	450
length_of_phenophase_4	Length of the 4th phenophase expressed in GDDs. In the case of maize this refers to late vegetative growth (this phenophase ends with anthesis).	240	450
length_of_phenophase_6	Length of the 6th phenophase expressed in GDDs. In the case of maize this refers to grain filling after anthesis.	850	1200
leaf_allocation_3	Fraction of total daily allocation that is associated with leaf growth in the 3rd phenophase.	0.4	0.5
root_allocation_3	Fraction of total daily allocation that is associated with root growth in the 3rd phenophase.	0.3	0.5
stem_allocation_3	Fraction of total daily allocation that is associated with stem growth in the 3rd phenophase.	0.2	0.5
leaf_allocation_4	Fraction of total daily allocation that is associated with leaf growth in the 4th phenophase.	0.2	0.5
root_allocation_4	Fraction of total daily allocation that is associated with root growth in the 4th phenophase.	0.2	0.4
stem_allocation_4	Fraction of total daily allocation that is associated with stem growth in the 4th phenophase.	0.3	0.5
root_allocation_6	Fraction of total daily allocation that is associated with root growth in the 6th phenophase.	0.05	0.15
fruit_allocation_6	Fraction of total daily allocation that is associated with grain filling in the 6th phenophase.	0.5	0.8
stem_allocation_6	Fraction of total daily allocation that is associated with stem growth in the 6th phenophase.	0.1	0.4
maxlifetime_3	Maximum, genetically determined lifetime of new leaf tissue in the 3rd phenophase expressed in GDDs.	500	1300
maxlifetime_4	Maximum, genetically determined lifetime of new leaf tissue in the 4th phenophase expressed in growing-degree-days.	500	1300

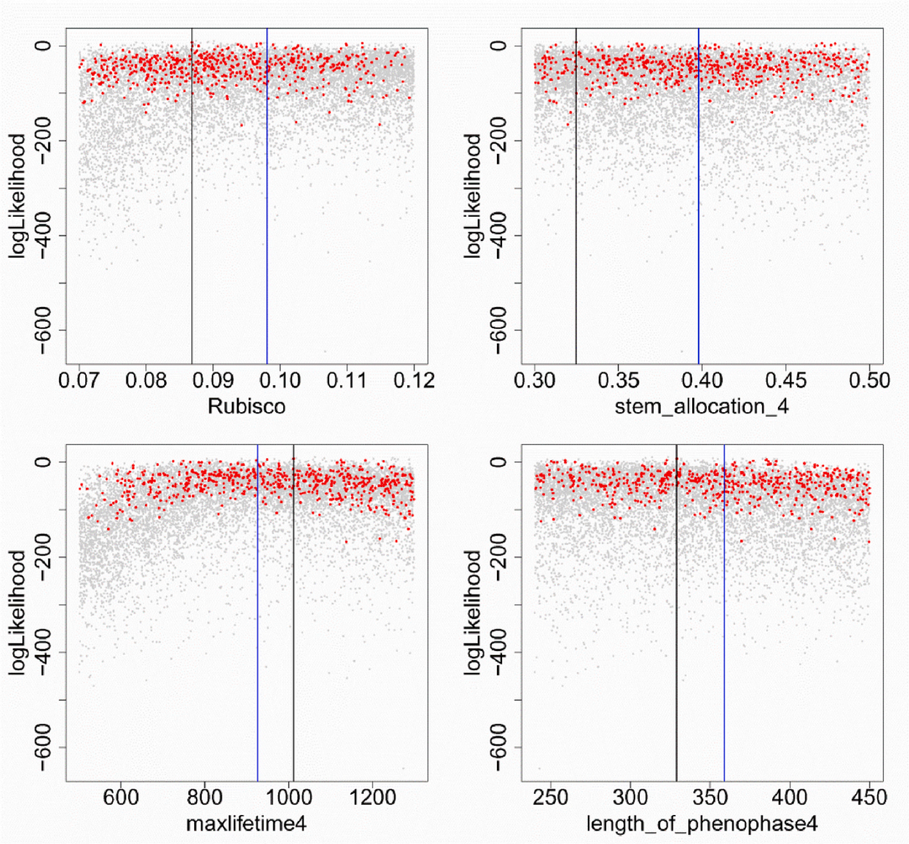

Fig. 2. Selected dotty plots from the optimization after the first iteration step (all simulation results are plotted including behavioral and non-behavioral ones). Grey dots represent the likelihood values for all simulations (marginal distributions), while red dots show the likelihood values that are associated with constrained simulations. Black vertical line represents the parameter value based on the maximum likelihood estimator, while blue vertical line is associated with the median value of the behavioral parameters of the constrained simulations after the first iteration step.

parameters are related to allocation.

Similarly to other models that are characterized with a high number of parameters (e.g. Bilionis et al., 2015), it is impossible to optimize the model for all parameters. Instead, we used previous experience and the outcome of several previous model evaluations for the parameter selection and parameter setting. During previous model optimization efforts (see Methods) the selection of the most relevant parameter values were done partly using objective sensitivity analysis, and partly by manual parameter adjustment. Many parameters were fixed because of available observation data (like leaf, stem and fine root C:N ratios), and the results of experience at the German and USA experimental sites (maximum stomatal conductance, canopy light extinction coefficient, etc). The remaining parameters were found to be variable between the sites and the model showed sensitivity to the proper setting of the parameters (see Table 1 for a full list).

For the selected parameters, prior to model optimization the upper and lower bounds were set based on literature values (Stöckle and Nelson, 2013; White et al., 2000). Expert knowledge of the co-authors and previous experience was also used to set the intervals (Table 1). Note that the complete prior ecophysiological parameterization for the simulations of maize is presented in the Supplementary material.

For the practical implementation of the procedure with Biome-BGCMuSo at the Martonvásár LTFE site we used a uniform distribution as prior over convex polytope. We used a normal likelihood function (Eq. (8)) and we assumed that modelization uncertainties are negligible compared to observational uncertainties. We used 0.01 for the complexity based pruning factor for the DT as it was the default value in the rpart package (Therneau et al., 2022). The white box model was the decision tree with CART as described above.

We used 4 output-constraint functions (f) to evaluate feasible or infeasible simulations. The first constraint is related to the annual Harvest Index (HI) that is defined by the ratio of the final grain yield and total aboveground biomass at harvest (Goudriaan et al., 2001). The median of the simulated HI values is required to be in the range of [0.40, 0.55] defined based on several scientific publications (Hütsch and Schubert, 2018; Ion et al., 2015; Li et al., 2015, 2017; Liu et al., 2020). The median of the annual maximum Leaf Area Index ($\mathrm{LAI}_{\mathrm{max}}$) is requested to be between 2.7 and 5 m^2/m^2 that is an observation based setting for Hungary (see e.g. Pokovai and Fodor, 2019). The long term median value of the rooting depth at the beginning of the flowering phenophase should be larger than 1.4 m but should be less than 1.8 m (those values are based on expert knowledge). The median of anthesis days (expressed in day-of-year: DOY) should be between 180 and 190 that is a typical range for Hungary.

The output-constraint functions are defined by a simple algorithm based on the model outputs. For example, if annual LAImax is a vector containing annual maximum values of the Biome-BGCMuSo simulated LAI, f is defined as:

$$f(x) = \begin{cases} 1 & med(LAI_{max}) \in [2.7, 5] \\ 0 & otherwise \end{cases} \tag{9}$$

where med represents the median of the values from the simulated years during the point simulation.

Biome-BGCMuSo simulations were performed within the RBBGCMuso software environment (https://github.com/hollorol/RBBGCMuso). RBBGCMuso is an open source R package supporting the easy and user-friendly application of the model. We used the CIRM branch of RBBGCMuso for the case study presented in this paper. The output conditioning method was implemented using 10 iterative steps. In each step the automated algorithm performed post-processing of the decision trees using the method described above (i.e. the interval refinement was done automatically taking into account the overlaps in the calculated thresholds; Step 4 in Algorithm 1). 10 000 simulations were done for each iteration step. Standard R was used mostly in the work (R Core Team, 2021) Additionally, the rpart package was used to construct the decision trees (Therneau et al., 2022).

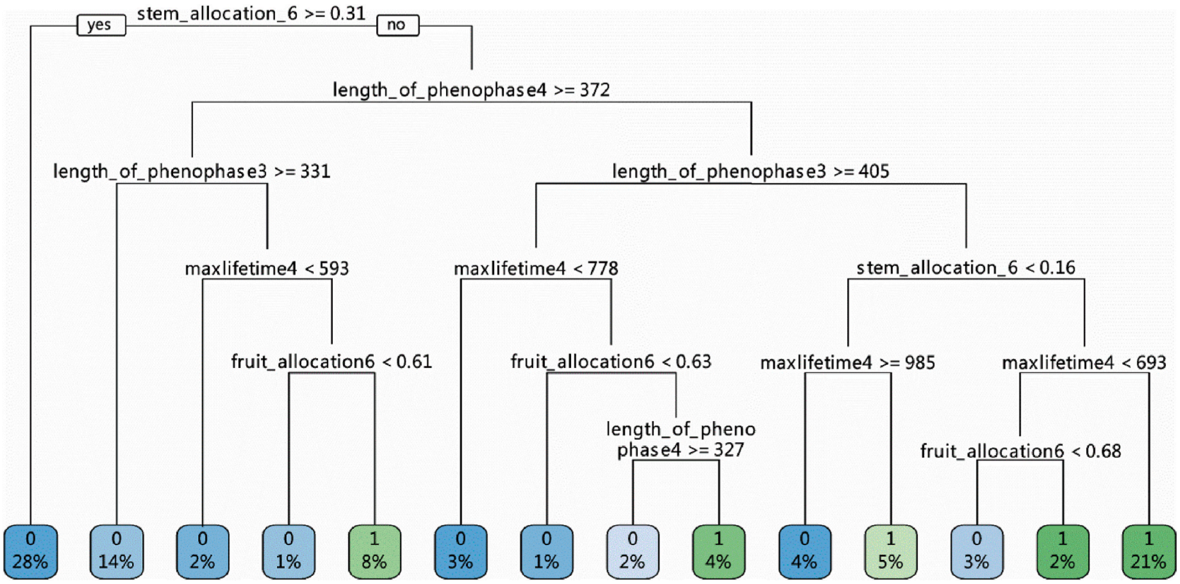

Fig. 3. Decision tree after the first iteration step based on the Harvest Index constraint. Note that the sum of the percentage values inside the leaf nodes is not 100% due to rounding.

In each iteration step the maximum likelihood parameter set was stored. A posteriori parameter intervals were estimated based on behavioral (top 5%) parameter values that matched the predefined conditioning. During the first 9 iteration steps parameter interval reduction was performed based on the decision tree update algorithm. GLUE-based a posteriori interval reduction was considered only in the final iteration step. Optimum parameter set was calculated as the median of the behavioral parameters. Here we refer to these intervals and optimum parameter set as GLUE-based, constrained intervals.

Optimized model performance was evaluated using the square of the linear correlation coefficient (R^2), bias (systematic error), root mean square error (RMSE) and Nash-Sutcliffe modelling efficiency (ME) (Ma et al., 2011; Sándor et al., 2016). The performance indicators were calculated based on the observed and the simulated maize yield time series. The optimized model was evaluated based on the training dataset, and also on the NUTS3 level simulation using independent data. In this latter case mean annual maize yield was calculated from the model simulations and the final time series was evaluated against the observed census data. We also quantified the percent of change in the final (GLUE) parameter intervals relative to the a priori intervals.

3. Results

3.1. GLUE and post-processing via the DTs

Fig. 2 shows selected dotty plots from the first iteration step. From the 20 studied parameters (Table 1) we selected 4 that represent typical patterns (characteristic to all cases) from the final (10th) iteration step (see below). Supplementary material Fig. S1 shows the complete set of dotty plots for the first step.

The most prominent feature of the dotty plots is equifinality based on the grey dots that represent all simulations. This is the case for almost all other parameters that are presented in Fig. S1. There are a few exceptions where some pattern can be recognized on the graphs (maxlifetime4, leaf_allocation_3, leaf_allocation_4, stem_allocation_3, and Rubisco to some extent; Fig. S1), but the distribution of the grey dots support only a slight interval reduction (for e.g. stem_allocation_3, leaf_allocation_4). For other parameters such as maxlifetime4, even if some pattern can be recognized in the dotty plots, posterior intervals based on the behavioral grey dots will not result in parameter interval reduction. Note that in a typical Bayesian calibration this is the final stage of the optimization which is clearly not satisfying and unsuccessful in terms of pursuing an interval reduction.

After we decided to check the consistency of the results based on the predefined constraints (using HI, LAI_{max}, anthesis date and rooting depth), the overall picture changed (Fig. 2, Fig. S1; red dots). However, likely given the large degree of freedom of the optimization, after the first iteration only 596 simulations were feasible, which is clearly a low success rate ($c_{r} = 5.96\%$). With the output constraint filtering the dotty plots still show equifinality for most of the cases. There are some exceptions like Rubisco, maxlifetime4, leaf_allocation_3, root_allocation_3, stem_allocation_3, leaf_allocation_4, root_allocation_4, fruit_allocation_6 and stem_allocation_6 for which the algorithm based on constraints suggests that they should have narrower ranges (Fig. 2; Fig. S1). For example, in the case of Rubisco the parameter range ~ [0.7,0.11] seems to be reasonable based on the red dots. We would like to stress here again that in a typical optimization exercise the user neglects some or all above mentioned constraints and conclusions are drawn based on the grey dots only. In this sense information extracted from the feasible simulations is already one step forward.

One can recognize that because of the small c_r we do not have a sufficient number of feasible simulations to trust the goodness of the GLUE optimum values and the GLUE uncertainty ranges. In order to increase c_r, further refinement of the parameter intervals is a reasonable next step. The constructed decision trees provide information about the possible relationships between the parameters and the feasible/infeasible simulations, thus they can be useful for updating the parameter intervals. Given the fact that we have 4 output constraints, 4 DTs were constructed.

Fig. 3 shows the DT that was constructed based on the first constraint (HI) after iteration step #1. The top level of the DT is called the root node (in the case of Fig. 3 this is represented by stem_allocation_6). This top level always shows the most important parameter that affects the feasibility of the simulation (in this case in terms of HI). The lower part of the DT is divided into different layers (or levels) where internal nodes are located. The importance of the parameters associated with the different internal nodes is decreasing with the layer number (i.e. the importance decreases from top to bottom). Internal nodes represent additional decisions revealing the parameter value that splits the parameter range into two parts depending on the feasible/infeasible character of given simulations. For example, at layer 2 the cutting threshold value for the length_of_phenophase_4 parameter is 372. If the

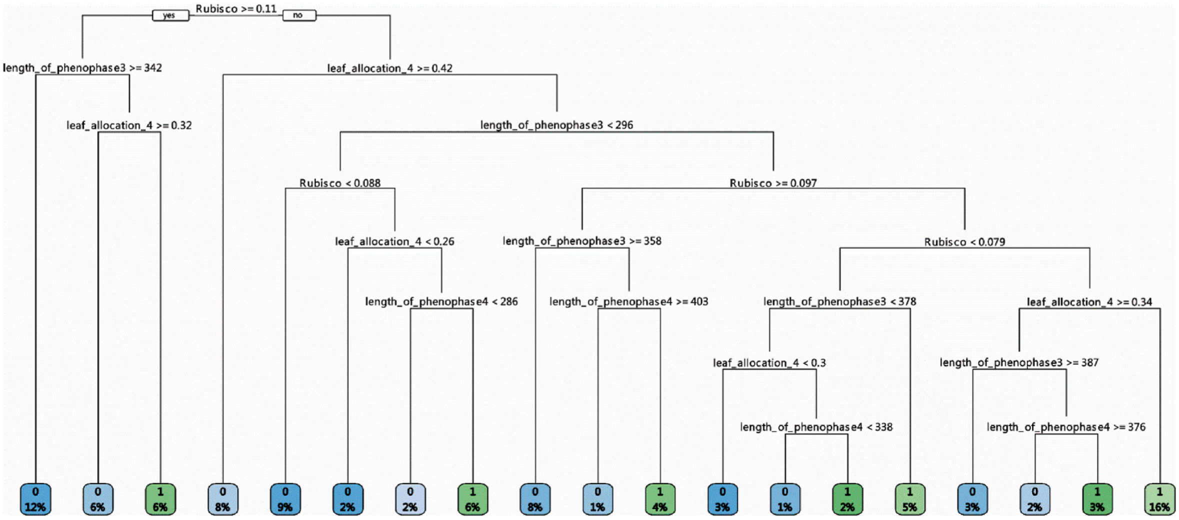
Rubisco >= 0.11
yes
no
length_of_phenophase3 >= 342
leaf_allocation_4 >= 0.42
leaf_allocation_4 >= 0.32
length_of_phenophase3 < 296
Rubisco < 0.088
Rubisco >= 0.097
leaf_allocation_4 < 0.26
length_of_phenophase3 >= 358
Rubisco < 0.079
length_of_phenophase4 < 286
length_of_phenophase4 >= 403
length_of_phenophase3 < 378
leaf_allocation_4 >= 0.34
leaf_allocation_4 < 0.3
length_of_phenophase3 >= 387
length_of_phenophase4 < 338
length_of_phenophase4 >= 376
0 12%
0 6%
1 6%
0 8%
0 9%
0 2%
0 2%
1 6%
0 8%
0 1%
1 4%
0 3%
0 1%
1 2%
1 5%
0 3%
0 2%
1 3%
1 16%

Fig. 4. Decision tree after the first iteration step based on the LAI_{max} constraint. Note that the sum of the percentage values inside the leaf nodes is not 100% due to rounding.

parameter is less than 372 then we can reach the right branch with the rest of the levels. At the bottom of the DT the leaf nodes are located. Leaf nodes represent the results of the classification due to the homogeneity of the leaf nodes. In Fig. 3 blue leaf nodes (i.e. rounded squares at the bottom marked with 0 which is the result of $f \circ \mathscr{M}$) indicate infeasible simulations, while green leaf nodes represent feasible simulations (marked with 1). The percentage values inside the leaf nodes show the fraction of simulations that is associated with a given branch.

According to Fig. 3, due to the applied HI constraint 58% of the sampled parameter combinations were infeasible (sum of the percentages in the blue boxes). As it was explained in section 2.4.2. in our approach we always focus on the leaf node with the highest percentage of feasible simulations. In this sense the DT suggests that the most important parameters associated with the HI constraint in the decreasing order are stem_allocation_6, length_of_phenophase_4, length_of_phenophase_3, and maxlitefime_4 (this is the path from the top to the leaf node with the highest % after excluding the repeated occurrence of the parameters in the lower layers of DT). Those parameters indirectly affect grain allocation in the model, and thus HI. It is somewhat surprising that fruit_allocation_6 parameter is not included in the DT in this path (but it is included in other paths leading to other leaf nodes). Maxlifetime4 affects leaf senescence dynamics prior to and during grain filling thus interacts with HI. The lengths of phenophases 3 and 4 affect the leaf dynamics that clearly influence assimilation thus grain allocation during the 6th phenophase. The information gained from the DT is essentially useful and provides insights into the complex process of plant growth and final yield that is implemented in Biome-BGCMuSo.

Based on the decisions across the path from top to the leaf node associated with the largest success rate (21%, rightmost leaf node) we can extract the thresholds and update the original parameter intervals (see Table 1). Stem allocation in the 6th phenophase should be less than 0.31 and greater than 0.16 (the original interval was [0.1,0.4]; see Table 1). The length of the 3rd phenophase should be smaller than 405, and the length of the 4th phenophase should be smaller than 372. Maxlifetime4 should be set larger than 693.

Note that at any time of the procedure a user might select a different leaf node with a lower success rate if the expert knowledge supports an alternative choice. In this case manual adjustment might be needed in the parameter ranges and the iteration should be restarted using the adjusted intervals.

Fig. 4 shows the DT for the LAI_{max} constraint after iteration step #1.

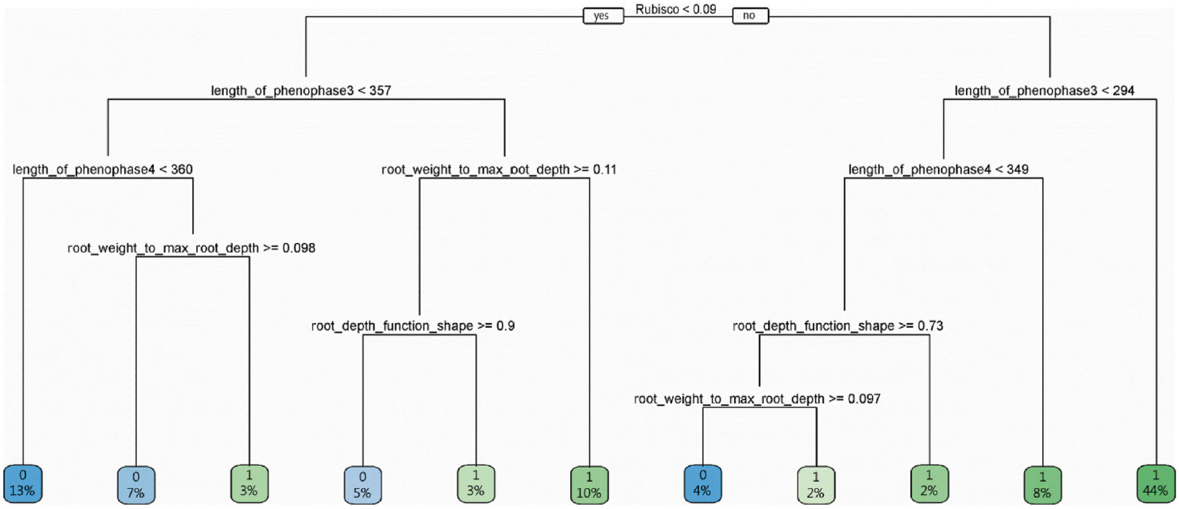
Rubisco < 0.09
yes
no
length_of_phenophase3 < 357
length_of_phenophase3 < 294
length_of_phenophase4 < 360
root_weight_to_max_root_depth >= 0.11
length_of_phenophase4 < 349
root_weight_to_max_root_depth >= 0.098
root_depth_function_shape >= 0.9
root_depth_function_shape >= 0.73
root_weight_to_max_root_depth >= 0.097
0 13%
0 7%
1 3%
0 5%
1 3%
1 10%
0 4%
1 2%
1 2%
1 8%
1 44%

Fig. 5. Decision tree after the first iteration step based on the root depth constraint. Note that the sum of the percentage values inside the leaf nodes is not 100% due to rounding.

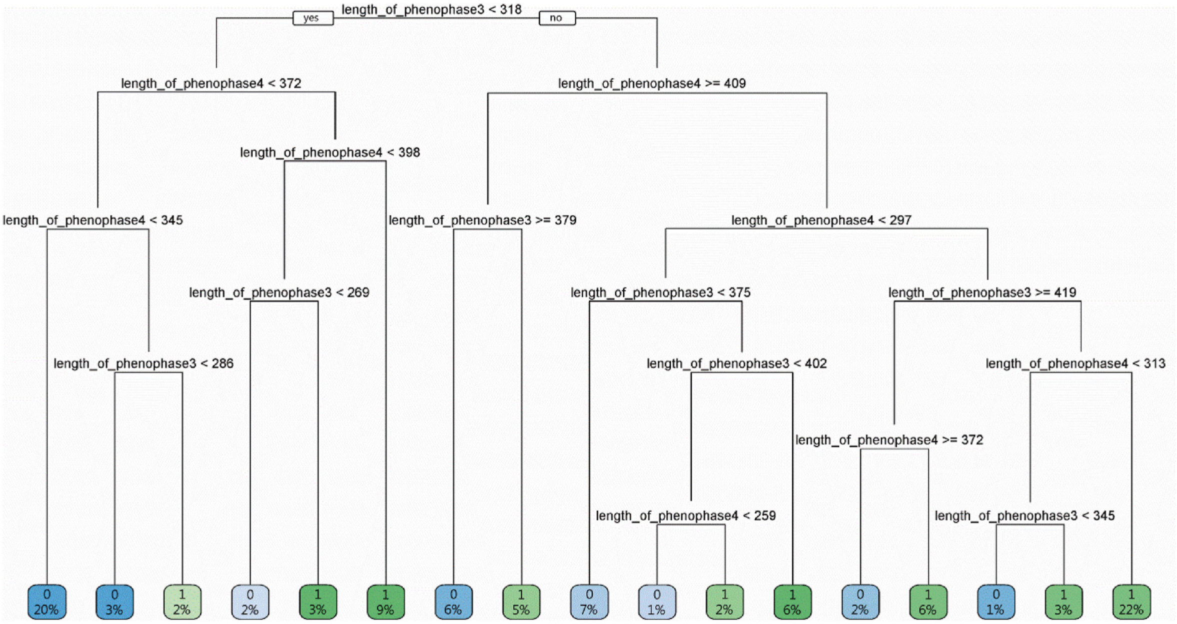
length_of_phenophase3 < 318
yes
no
length_of_phenophase4 < 372
length_of_phenophase4 >= 409
length_of_phenophase4 < 398
length_of_phenophase4 < 345
length_of_phenophase3 >= 379
length_of_phenophase4 < 297
length_of_phenophase3 < 269
length_of_phenophase3 < 375
length_of_phenophase3 >= 419
length_of_phenophase3 < 286
length_of_phenophase3 < 402
length_of_phenophase4 < 313
length_of_phenophase4 >= 372
length_of_phenophase4 < 259
length_of_phenophase3 < 345
0 20%
0 3%
1 2%
0 2%
1 3%
1 9%
0 6%
1 5%
0 7%
0 1%
1 2%
1 6%
0 2%
1 6%
0 1%
1 3%
1 22%

Fig. 6. Decision tree representing the first iteration step based on the anthesis date constraint. Note that the sum of the percentage values inside the leaf nodes is not 100% due to rounding.

The figure shows that 58% of the sampled parameter combinations is infeasible. The tree suggests that the most important parameters (in a decreasing order) are Rubisco, leaf_allocation_4 and length_of_phenophase3. At lower layers Rubisco and leaf_allocation_4 appear again. As Rubisco ultimately controls the maximum photosynthesis rate (White et al., 2000), its importance is straightforward in terms of leaf development. The role of leaf allocation in the 4th phenophase (which determines the peak LAI) is also clear and easily interpretable. The length of the 3rd phenophase is less intuitive but it is reasonable since it affects the initial condition of leaf development in the 4th phenophase when LAI reaches its maximum.

Using the DT and the path to the rightmost leaf node (that is associated with the highest success rate with 16%) we can set new intervals for the parameters. Based on all decision nodes from the DT in Fig. 4, Rubisco should be less than 0.097 and greater than 0.079, leaf_allocation_4 should be less than 0.34, and length_of_phenophase3 should be larger than 296. Note that the DT presented in Fig. 3 already set a new upper limit for length_of_phenophase3 which is further refined here.

Fig. 5 presents the DT for the rooting depth constraint based on the results from the 1st iteration step. In this case only 29% of the sampled parameter combinations were infeasible. The tree suggests that the most important related parameters are Rubisco and length_of_phenophase_3. This is reasonable considering the determinant role of Rubisco in terms of overall productivity, and considering the importance of the 3rd phenophase in terms of root establishment.

Rubisco interval can be further refined here (it should be larger than 0.09 and according to the previous tree in Fig. 4 it should be less than 0.097). Length_of_phenophase3 should be greater than 294 which is in fact not used at this stage as it was already set to be larger than 296 (Fig. 4).

Fig. 6 presents the DT for the last, anthesis date based constraint. Using the anthesis date DT 42% of the sampled parameter combinations turned out to be infeasible. The tree suggests that the most important parameters are the length of phenophases 3 and 4. This is in perfect agreement with the expectations as the anthesis date is driven by the length of the previous phenophases expressed in GDD (note that the length of the first two phenophases is fixed in the simulations).

The DT provides guidelines for updating the two parameters. According to the path leading to the rightmost leaf node length_of_phenophase3 should be set larger than 318, and length_of_phenophase4 should be larger than 313. These new settings further constrain these parameters as they were already refined to some

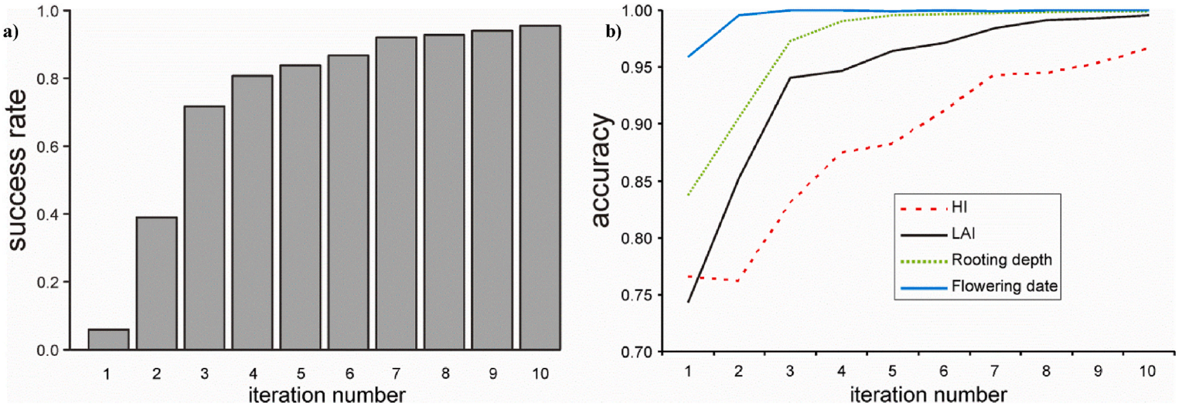

Fig. 7. a) Success rate as the function of the number of iteration steps based on the automated workflow. b) Accuracy of the DTs as the function of the iteration number.

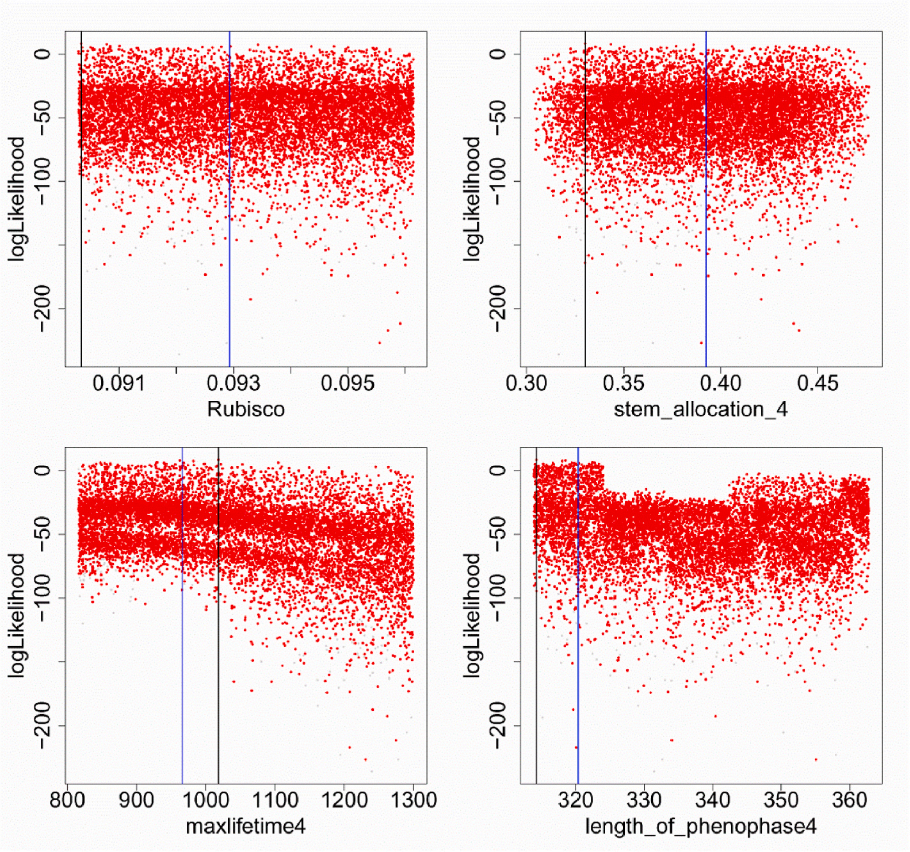

Fig. 8. Selected dotty plots from the optimization after iteration step 10. The meaning of the symbols and the vertical lines is the same as in Fig. 2.

extent.

At this stage the most important recognition is the usefulness of the DTs that can be used for manual updating of the parameter intervals. This kind of information was “hidden” so far as the marginal distributions did not reveal parameter interval reduction possibilities. After performing the DT analyses of individual constraints, the conditions from individual DT are combined, and the parameter intervals are modified.

3.2. Results of the iterations and automatic interpretation of the DTs

One might recognize at this point that the new interval settings might be used as the new prior of another Monte Carlo-based GLUE experiment. As it was described above, a custom procedure was developed to automatically interpret the DTs by finding the path to the leaf node that contains the highest percentage of feasible simulations. Based on this method the above described interval refinement became unattended. If the next iterations turn out to provide better success rates, and if the parameter interval reduction can be further refined, the procedure can be transformed into a multi-step, iterative method.

In our case, after the first iteration step another nine were performed. During the iterations the success rate (Fig. 7a) increased monotonically indicating that the introduced algorithm worked well on improving the success rate. As Fig. 7b shows, the white-box approximation was also correct, because on average the approximation’s accuracy monotonically increased, although it can be attributed to the large success rate after step 4.

Supplementary material Fig. S2 shows the complete set of dotty plots for the 10th iteration step. Fig. 8 shows selected dotty plots that represent typical patterns from the final step. The graph is in accordance with Fig. 7 revealing that almost all simulations were feasible at this stage (grey dots are hardly detectable). Although most of the parameters still show equifinality, the parameter intervals are considerably smaller than after the first step (Fig. 2; Fig. S1). Some of the parameters provide well detectable optimum (e.g. max_lifetime_4 at Fig. 8) with typical parameter distribution. Other parameters show well-bounded points like in case of stem_allocation_4. Parameters controlling the length of the phenophases 3 and 4 show an unusual distribution that is the clear consequence of the cutoff values defined by the DTs (see above). In those plots the behavioral parameters localize the optimum value.

After the 10th iteration step almost all sampled parameters were feasible (95.45%, i.e. 9545 iterations out of 10 000). It means that almost all simulations satisfied the predefined constraints thus provide results according to the expectations of the user. Given the large number of successful and meaningful simulations the high sample number clearly improves the confidence of the user about the statistical properties of the results (most of all the optimum and the uncertainty ranges).

Considering the optimum values of the parameters (represented by vertical lines in Fig. 8) the maximum likelihood values typically differed from those calculated from the behavioral data (i.e. GLUE median) similarly to step 1 (Fig. 2, Fig. S1).

Fig. 9 shows the summary of the inversion in the form of a special plot type that is referred to as “kitchen sink plot”. The figure provides easily interpretable information about the multiple-step iteration in terms of DT-based parameter interval reduction as a function of iteration step number, and it also shows the position of the maximum likelihood estimation relative to the parameter ranges. Note that in the 10th iteration step the GLUE-based interval reduction was also considered. The GLUE-based optimum is not indicated but this can be approximated by the mid-point of the actual interval.

Fig. 9 clearly shows that parameter intervals were significantly reduced in many cases. For some parameters such as Rubisco and length_of_phenophase4 interval reduction was more profound in the first

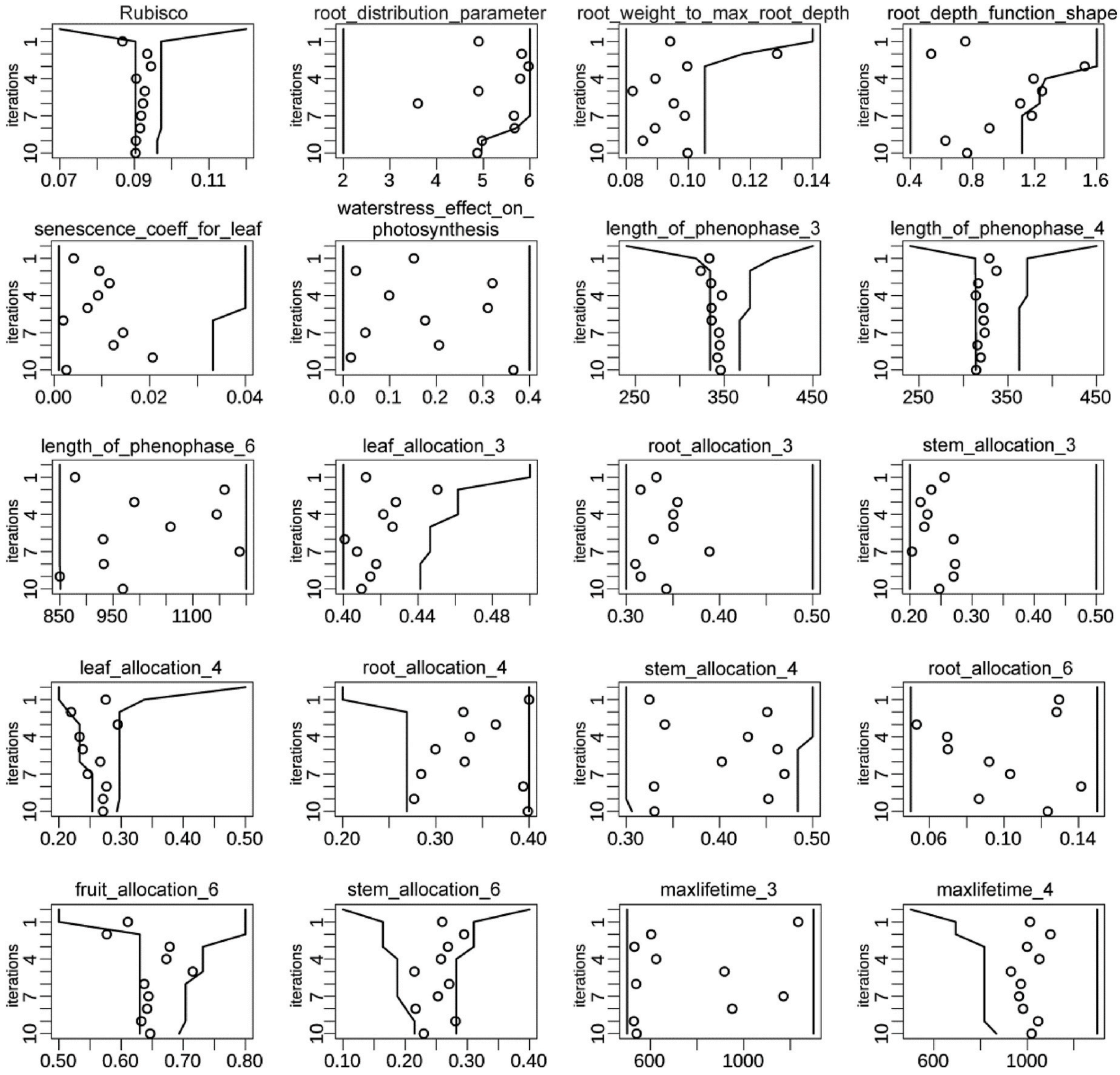

Fig. 9. "Kitchen sink plots" showing the performance of the introduced novel method. Lines indicate the lower and upper bounds for a given parameter and their changes between successive iteration steps, while circles indicate ML estimation results for the given iteration step. The abbreviations are defined in Table 1.

iteration step. For other parameters such as root_depth_function_shape, leaf_allocation_3 and stem_allocation_6 the reduction was gradual and not necessarily associated with one iteration step. This indicates the utility of the multiple-step approach. In case of some parameters the prior interval remained unchanged (e.g. root_allocation_3, stem_allocation_3). The plot also shows that in some cases the ML value was out of the final parameter interval (i.e. in step 10) that indicates an infeasible solution. As the first iteration step is the one which is performed in a usual Bayesian experiment, this clearly shows the „good results for wrong reasons" situation.

Table 2 summarizes the model optimization exercise providing information about the posterior parameter intervals, the parameter set representing the maximum likelihood estimation in the final step, and the GLUE-based parameter ranges. Average interval percentage change was 44% for the 20 studied parameters (the maximum was 88% associated with Rubisco).

Parameter range reduction did not occur for length_of_phenophase_6, root_allocation_3, stem_allocation_3, root_allocation_6 and maxlifetime_3. Stem and root allocation in the 3rd phenophase seem to be less determinant in terms of the final crop yield which is an interesting outcome. The interpretation of this parameter behavior from the 6th phenophase is easier. Estimation of the exact length of this phenophase might be impossible since after leaf senescence during grain filling the final crop yield cannot change anymore, so its value cannot be set by any observation or constraint. Root allocation in the 6th phenophase is small and seems to have no substantial effect on the final crop yield.

Table 2 suggests that additional constraints might support the reduction of the intervals in the parameters where 0% reduction is present. Note that in some cases this is not a problem as the other dependent parameters already set the value for these parameters (like in case of the allocation when the sum of the parameters must sum up to 1).

Overall, the parameter interval reduction have led to 42.3% decrease in the simulated yield uncertainty quantified by the mean of the annually calculated standard deviation of the modeling results based on 1000 simulations performed by Monte-Carlo based sampling from the original and reduced parameter ranges (Fig. S3 in the Supplementart material).

3.3. *Performance analysis on the calibration dataset and validation*

Fig. 10 shows the model results for the prior parameterization and for the GLUE-based, optimized parameter set. Uncertainty of the

Table 2
List of the optimized parameters with constrained intervals after 10 iterations. Maximum likelihood and GLUE-based optimized parameter values are provided as well. Percentage change of the interval is provided for the results of the final (10th) iteration step (see Fig. 5) relative to the prior range (c.f. Table 1).

Abbreviation	MIN	MAX	ML	GLUE	percentage change
Rubisco	0.09029	0.09615	0.0903	0.0929	88
root_distribution_parameter	2	4.966	4.8722	3.629	26
root_weight_to_max_root_depth	0.08	0.1053	0.09973	0.0920	58
root_depth_function_shape	0.4	1.118	0.7649	0.7583	40
senescence_coeff_for_leaf	0.001	0.03323	0.00255	0.01469	17
water stress_effect_on photosynthesis	0	0.4	0.3654	0.1886	43
length_of_phenophase_3	334.2	367.4	345.96	342.63	84
length_of_phenophase_4	313.9	362.8	314.279	320.392	77
length_of_phenophase_6	851.3	1200	968.46	1007.82	0
leaf_allocation_3	0.4	0.4412	0.4096	0.417	59
root_allocation_3	0.3	0.5	0.3429	0.339	0
stem_allocation_3	0.2	0.5	0.2473	0.244	0
leaf_allocation_4	0.2537	0.2932	0.2709	0.273	87
root_allocation_4	0.2689	0.4	0.3987	0.334	34
stem_allocation_4	0.306	0.4841	0.3302	0.393	11
root_allocation_6	0.05	0.15	0.1234	0.095	0
fruit_allocation_6	0.6299	0.6933	0.64670	0.657	79
stem_allocation_6	0.2149	0.2824	0.22983	0.248	78
maxlifetime_3	500	1300	540.46	930.70	0
maxlifetime_4	866.8	1300	1018.27	965.83	46

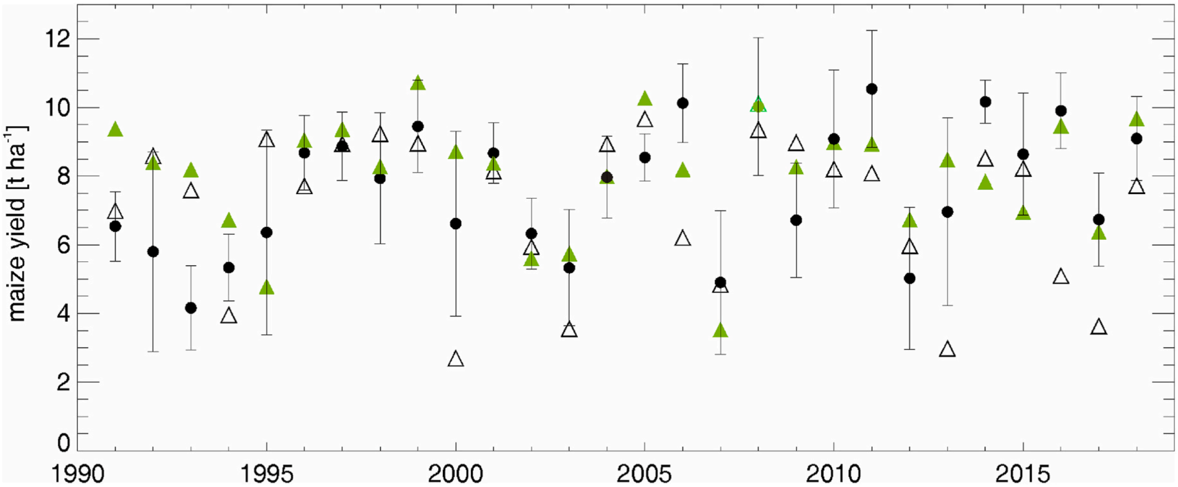

Fig. 10. Time series of the mean observed maize yield (filled circles; uncertainty is±one standard deviation), the a priori simulated (empty triangles) and the optimized (GLUE-based; green filled triangles) maize yield at Martonvásár from 1991 to 2018.

Table 3
Statistical evaluation of the model performance after individual iteration steps using the long-term (1991–2018) field experiment data of maize yield from Martonvásár.

Parameter estimation	ML				GLUE			
	R^2	RMSE	bias	ME	R^2	RMSE	bias	ME
a priori	0.16	4.752	−4.448	−5.793	0.16	4.752	−4.448	−5.793
Step 1	0.37	1.548	0.486	0.279	0.07	2.209	0.324	−0.468
Step 2	0.41	1.411	−0.003	0.401	0.28	1.675	0.051	0.156
Step 3	0.51	1.326	0.176	0.471	0.02	2.408	−0.063	−0.744
Step 4	0.44	1.458	0.449	0.360	0.33	1.689	0.268	0.142
Step 5	0.45	1.388	0.043	0.421	0.33	1.678	0.158	0.153
Step 6	0.36	1.552	0.478	0.275	0.41	1.552	0.269	0.276
Step 7	0.41	1.461	0.167	0.358	0.38	1.573	0.377	0.256
Step 8	0.46	1.391	0.002	0.418	0.38	1.548	0.270	0.280
Step 9	0.40	1.465	−0.045	0.354	0.39	1.558	0.292	0.270
Step 10	0.48	1.358	−0.102	0.446	0.38	1.585	0.395	0.244

observations is high given that the maize yield dataset is a composite of many small experimental plots (see Methods). The figure indicates that in many cases the optimized model estimated yield within the uncertainty ranges. Improvements were quantified by using statistical indicators.

Table 3 shows the results of the statistical evaluation of the simulations. Here we include performance metrics from the maximum likelihood simulations as well. The table shows considerable improvements of

Table 4
Comparison of error metrics for the different maize yield simulations. Martonvásár represents the point simulation (training) dataset, while Fejér county means the independent, census data based model evaluation using the different parameterizations. For Fejér county the final parameter set was used. See text for details.

	R^2	RMSE	bias	ME
a priori at Martonvásár	0.16	4.752	−4.45	−5.793
a priori at Fejér county	0.15	3.500	−3.08	−2.658
step 10 ML at Martonvásár	0.48	1.358	−0.10	0.446
step 10 GLUE at Martonvásár	0.38	1.585	0.39	0.244
step 10 ML at Fejér county	0.37	1.986	1.28	−0.109
step 10 GLUE at Fejér county	0.48	2.028	1.49	−0.157

the quality of the simulations in the consecutive iterations steps. The explained variance was typically higher for the maximum likelihood parameterization than for GLUE. For maximum likelihood R^2 substantially increased already after 1st iteration while this was achieved only after the 6th iteration for GLUE. RMSE was lower for the maximum likelihood than for GLUE (exception is step 6). The bias was variable but basically became close to 0 in the case of maximum likelihood while remained positive for GLUE. ME was better in the case of the maximum likelihood parameterization than in the case of GLUE. In summary, all parameter values show better performance in the case of the maximum likelihood parameterization which might indicate overfitting.

Table 4 shows the performance metrics of the model output for maize yield at Martonvásár and for the independent validation experiment (NUTS3 level model simulation). Both for the calibration set (Martonvásár) and the validation set the optimized simulation results were significantly closer to the observations than the a priori simulations. It means that the calibrated model is more appropriate to apply to the NUTS3 level independent dataset.

Maximum likelihood parameterization from the 10th step overperformed the GLUE-based simulations in terms of bias and ME, while the difference between the RMSE values was small. In terms of R^2 the GLUE-based method performed better. The explained variance (48%) was in fact higher here than in the case of the training dataset for GLUE. Note that ME was negative in both cases because the magnitude of the yield was different at Martonvásár and at the county-level likely due to the different N fertilization level and agrotechnology. The better performance of GLUE in the validation experiment in terms of R^2 indicates that maximum likelihood was associated with over-fitting and the GLUE-based method might be more feasible. Note that at this stage the modeler might perform a hybrid parameterization using the maximum likelihood values for some parameters that are associated with small or zero interval reduction, and GLUE-based, more constrained parameters in other cases. Maximum likelihood values for some parameters might be informative if there is consensus in their values across the multiple iterations steps.

Note that the low explained variance does not necessary mean bad simulation. In our case the likelihood (optimum) function was normal so during the training procedure the main goal was to minimize the error and not to maximize R^2.

3.4. Limitations and outlook

Like all optimization methods, CIRM also has limitations. The method assumes that the DT white-box model is an adequate approximation to the black box model ($\mathscr{M}\circ f$). Similarly to any machine learning classifiers, DTs have problems with unbalanced datasets, because these are heavily biased towards the majority classes (Hoens and Chawla, 2013). However, quantifying the goodness of the approximation can be accomplished easily by checking the accuracy of the DT while simultaneously checking the success rate. If the success rate monotonically increases while the accuracy is above a predefined threshold (e.g., 0.7 is considered high enough), the DT approximation is considered adequate through the updating procedure. Another problem is that the DTs are prone to overfitting. Traditionally, the solution for the overfitting is the application of ensemble methods such as Random Forest, or AdaBoost (James et al., 2013). However, applying these solutions may result in loss of interpretability and might complicate the interval update algorithm. Additional research should focus on this issue in the future. Another direction of development can focus on the application of other performance metrics (such as recall that uses TP/(TP + FN)) for the decision trees that are more suitable for unbalanced datasets. In this paper, DTs are trained on the full available dataset because the success rate calculation and the accuracy value in the consecutive iteration can be considered as a simple validation.

In this study the introduction of the CIRM method was done using uniform priors coupled with the widely used GLUE probabilistic method. CIRM can be used with other priors and also with other probabilistic model optimization methods. In such cases the prior update rule has to be defined based on the DTs. For example, for normal prior distributions $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ parameters can be determined in a way that the 0.95 Highest Density Interval's endpoints are generated by the DT update rule as defined earlier. This method is applicable for every two-parameter unimodal distribution function (e.g Beta distribution).

It is also important to emphasize that there are alternatives to the interval update algorithm while processing the DT (see section 2.4.2.). The algorithm presented here selects the maximal volume inner hyperrectangle inscribed into the resulting polytope described by the DT (Fig. 1). Alternatively, the bounding hyperrectangle could also be used. Further research is needed to compare these two simple alternatives and see their effect on the results.

In this study the demonstration of the introduced new method was done at a single site with few available observations. In spite of the poor situation, the method could provide useful results which means potential in other similar, or more data-rich cases. Most importantly, the method has to be tested at experimental sites equipped with eddy covariance measurements and ancillary measurements.

Data retrieved from the TRY database can be included in the model optimization with predetermined ranges for some plant traits (Kattge et al., 2011). According to the introduced method it is possible to handle observations as additional constraints, together with another observation data stream that is used for the construction of the likelihood function.

4. Concluding remarks

In this study we presented a novel approach for the inversion of process-based models that goes beyond the traditional probabilistic methods. Although the basic aim is unchanged (i.e. constraining parameter uncertainty), the method is markedly different and can be best described as a combination of the traditional probabilistic methods and the application of an interpretable machine learning method.

Up to the knowledge of the authors there is no similar approach published in the literature. Although the concept of "reality constraints" as part of the model optimization is already introduced in the literature, the previously proposed methods deal exclusively with the input model parameter constraints, and no conditioning is present regarding the output data streams. Note that input data conditioning is an implicit feature of our procedure and is implemented in the RBBGCMuso package that was used to execute the optimization.

The so-called Bayesian filtering technique shows some similarity to our method This method implements conditioning of the output streams, but as their convergence speed is proportional to the ratio of realistic/all simulations, they can be very slow. Additionally, this method requires in-depth knowledge from the user regarding the underlying (inter)dependencies of the parameter space.

Interpretability is a remarkable advantage of the CIRM method. In this regard, the output conditioning introduced in this study is a novel technique. Another advantage of the proposed method is that it has two modes: a manual and an automatic one. In the manual mode the user can examine the relationships represented by the constructed DTs and perform the interval refinement accordingly, or the proposed refinement can be accepted or modified based on additional scientific knowledge. The automatic interval refinement needs no supervision. In this way this method could be a potential candidate for researchers who used the trial and error approach for model inversion so far. The new method requires minimal extra knowledge about the technical implementation of the method from the user and as the method is fully automatic inexperienced researchers might still use it successfully concentrating mainly on the scientific questions.

The focus of the study was a low-data situation when the model had to be optimized against maize yield data with relatively large uncertainty. The study demonstrated that the traditional probabilistic method (GLUE) resulted in unconstrained parameter intervals suggesting that the low-data situation leads to a large uncertainty of optimized parameter values. This problem was solved by applying the proposed CIRM method that eventually led to successful interval reduction with almost 100% realistic simulation score (before the calibration only 6% of the simulations were acceptable).

Despite all of the achievements and proposed solutions in the field of model optimization, reality is sobering. Most of the scientists still use trial-and-error model optimization (Wallach et al., 2021), and the minority who preferred some probabilistic methods typically use only prior knowledge. There is a clear and well-recognized need for simple, easy-to-use and interpretable software solutions that can be used for parameter estimation of a wide array of process-based models. CIRM might represent a remarkable major step forward to support improved model optimization and application. Given the fact that CIRM is a model-independent method, it can be easily implemented in any modelling environment.

Software availability

The RBBGCMuso package that is used in the study is available at GitHub: https://github.com/hollorol/RBBGCMuso/tree/CIRM. Biome-BGCMuSo is available at the website of the model: http://nimbus.elte.hu/bbgc/

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Observation data used in the study is available at GitHub within the CIRM branch of the RBBGCMuso software package: https://github.com/hollorol/RBBGCMuso/blob/CIRM/docs/CIRM/Martonvasar_maize.obs. NUTS3 level census data on maize yield is available at the website of the Hungarian Central Statistical Office: https://www.ksh.hu/docs/hun/xstadat/xstadat_eves/i_omn013a.html. Climatic data used in the study have been obtained from the FORESEE database and is available at GitHub: https://github.com/hollorol/RBBGCMuso/blob/CIRM/docs/CIRM/Martonvasar.wth.

Acknowledgements

The research was funded by the Széchenyi 2020 programme, the European Regional Development Fund and the Hungarian Government (GINOP-2.3.2-15-2016-00028), by the National Multidisciplinary Laboratory for Climate Change (RRF-2.3.1-21-2022-00014) project, and by grant "Advanced research supporting the forestry and wood-processing sector's adaptation to global change and the 4th industrial revolution", No. CZ.02.1.01/0.0/0.0/16_019/0000803 financed by OP RDE. Also supported by the Integrated Infrastructure Operational Programme funded by the ERDF (project "Scientific support of climate change adaptation in agriculture and mitigation of soil degradation", grant no. ITMS2014 + 313011W580). The Klingenberg EC measurements have been funded by EU project CARBOEUROPE-IP (GOCE-CT-2003-505572), BMBF for ICOS-D implementation (1LK1101A, 1LK1101B) and Freistaat Sachsen (1–0230.00/68/192–2016/118972, 1–0452/147/158, 65–0456/49/10, 65–0456/46/28). We thank Andy Suyker, Steve Evett and Bruce Kimball for providing us the Bushland and Mead experimental data for benchmarking the model.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.envsoft.2022.105556.

References

Angulo, C., Rötter, R., Lock, R., Enders, A., Fronzek, S., Ewert, F., 2013. Implication of crop model calibration strategies for assessing regional impacts of climate change in Europe. Agric. For. Meteorol. 170, 32–46.

Bellman, R., 1957. Dynamic Programming. Princeton University Press.

Beven, K., Binley, A., 2014. GLUE: 20 years on. Hydrol. Process. 28, 5897–5918. https://doi.org/10.1002/hyp.10082.

Beven, K., Freer, J., 2001. Equifinality, data assimilation, and uncertainty estimation in mechanistic modelling of complex environmental systems using the GLUE methodology. J. Hydrol. 249, 11–29. https://doi.org/10.1016/S0022-1694(01)00421-8.

Bilionis, I., Drewniak, B.A., Constantinescu, E.M., 2015. Crop physiology calibration in the CLM. Geosci. Model Dev. (GMD) 8, 1071–1083.

Bloom, A.A., Williams, M., 2015. Constraining ecosystem carbon dynamics in a data-limited world: integrating ecological" common sense" in a model-data fusion framework. Biogeosciences 12, 1299–1315.

Braswell, B.H., Sacks, W.J., Linder, E., Schimel, D.S., 2005. Estimating diurnal to annual ecosystem parameters by synthesis of a carbon flux model with eddy covariance net ecosystem exchange observations. Global Change Biol. 11, 335–355.

Brunetti, G., Stumpp, C., Šimůnek, J., 2022. Balancing exploitation and exploration: a novel hybrid global-local optimization strategy for hydrological model calibration. Environ. Model. Software 150. https://doi.org/10.1016/j.envsoft.2022.105341.

Di Castro, F., Bertini, E., 2019. Surrogate decision tree visualization interpreting and visualizing black-box classification models with surrogate decision tree: 2019 Joint ACM IUI Workshops, ACMIUI-WS 2019. CEUR Workshop Proceedings 2327.

Dobor, L., Barcza, Z., Hlásny, T., Havasi, Á., Horváth, F., Ittzés, P., Bartholy, J., 2015. Bridging the gap between climate models and impact studies: the FORESEE Database. Geoscience Data Journal 2, 1–11.

Dumont, B., Leemans, V., Mansouri, M., Bodson, B., Destain, J.P., Destain, M.F., 2014. Parameter identification of the STICS crop model, using an accelerated formal MCMC approach. Environ. Model. Software 52, 121–135. https://doi.org/10.1016/j.envsoft.2013.10.022.

Fodor, N., Csathó, P., Árendás, T., Németh, T., 2011. New environment-friendly and cost-saving fertiliser recommendation system for supporting sustainable agriculture in Hungary and beyond. Journal of Central European Agriculture 12, 53–69.

Fodor, N., Pásztor, L., Szabó, B., Laborczi, A., Pokovai, K., Hidy, D., Hollós, R., Kristóf, E., Kis, A., Dobor, L., 2021. Input database related uncertainty of Biome-BGCMuSo agro-environmental model outputs. Int. J. Digit. Earth 14, 1582–1601.

Gelman, A., 1996. Bayesian model-building by pure thought: some principles and examples. Stat. Sin. 6, 215–232.

Gelman, A., Carlin, J.B., Stern, H.S., Rubin, D.B., 1995. Bayesian Data Analysis. Chapman and Hall/CRC.

Gelman, A., Yao, Y., 2020. Holes in bayesian statistics. J. Phys. G Nucl. Part. Phys. 48, 014002 https://doi.org/10.1088/1361-6471/abc3a5.

Gogna, A., Tayal, A., 2013. Metaheuristics: review and application. J. Exp. Theor. Artif. Intell. 25, 503–526. https://doi.org/10.1080/0952813X.2013.782347.

Goodfellow, I., Bengio, Y., Courville, A., 2017. Deep Learning (Adaptive Computation and Machine Learning Series). Cambridge Massachusetts, pp. 321–359.

Hararuk, O., Xia, J., Luo, Y., 2014. Evaluation and improvement of a global land model against soil carbon data using a Bayesian Markov chain Monte Carlo method. J. Geophys. Res.: Biogeosciences 119, 403–417.

Hartig, F., Calabrese, J.M., Reineking, B., Wiegand, T., Huth, A., 2011. Statistical inference for stochastic simulation models-theory and application. Ecol. Lett. 14, 816–827.

Her, Y., Chaubey, I., 2015. Impact of the numbers of observations and calibration parameters on equifinality, model performance, and output and parameter uncertainty. Hydrol. Process. 29, 4220–4237. https://doi.org/10.1002/hyp.10487.

Her, Y., Seong, C., 2018. Responses of hydrological model equifinality, uncertainty, and performance to multi-objective parameter calibration. J. Hydroinf. 20, 864–885. https://doi.org/10.2166/hydro.2018.108.

Hidy, D., Barcza, Z., Haszpra, L., Churkina, G., Pintér, K., Nagy, Z., 2012. Development of the Biome-BGC model for simulation of managed herbaceous ecosystems. Ecol. Model. 226, 99–119.

Hidy, D., Barcza, Z., Marjanovic, H., Sever, M.Z.O., Dobor, L., Gelybó, G., Fodor, N., Pintér, K., Churkina, G., Running, S., 2016. Terrestrial ecosystem process model Biome-BGCMuSo v4. 0: summary of improvements and new modeling possibilities. Geosci. Model Dev. (GMD) 9, 4405–4437. https://doi.org/10.5194/gmd-9-4405-2016.

Hidy, D., Barcza, Z., Hollós, R., Thornton, P., Running, S.W., Fodor, N., 2021. User's Guide for Biome-BGC MuSo 6.2.

Hidy, D., Barcza, Z., Hollós, R., Dobor, L., Ács, T., Zacháry, D., Filep, T., Pásztor, L., Incze, D., Dencső, M., 2022. Soil-related developments of the Biome-BGCMuSo v6. 2 terrestrial ecosystem model. Geosci. Model Dev. (GMD) 15, 2157–2181.

Hinne, M., Gronau, Q.F., van den Bergh, D., Wagenmakers, E.-J., 2020. A conceptual introduction to bayesian model averaging. Adv. Method. Pract. Psychol. Sci. 3, 200–215. https://doi.org/10.1177/2515245919898657.

Hoens, T.R., Chawla, N.V., 2013. Imbalanced datasets: from sampling to classifiers. In: Imbalanced Learning. John Wiley & Sons, Ltd, pp. 43–59. https://doi.org/10.1002/9781118646106.ch3.

Hütsch, B.W., Schubert, S., 2018. Maize harvest index and water use efficiency can be improved by inhibition of gibberellin biosynthesis. J Agro Crop Sci 204, 209–218.

Ion, V., Dicu, G., Dumbravă, M., Temocico, G., Alecu, I.N., Băşa, A.G., State, D., 2015. Harvest index at maize in different growing conditions. Romanian Biotechnological Letters 20, 10951.

IUSS Working Group, 2015. World Reference Base for Soil Resources 2014, Update 2015. International Soil Classification System for Naming Soils and Creating Legends for Soil Maps. World Soil Resources Reports No.106, Rome.

James, G., Witten, D., Hastie, T., Tibshirani, R., 2013. An Introduction to Statistical Learning. Springer.

Jarvis, H.J., 1989. A simple empirical model of root water uptake. Journal of Hydrology 107, 57–72.

Kattge, J., Díaz, S., Lavorel, S., Prentice, I.C., Leadley, P., Bönisch, G., Garnier, E., Westoby, M., Reich, P.B., Wright, I.J., Cornelissen, J.H.C., Violle, C., Harrison, S.P., Van Bodegom, P.M., Reichstein, M., Enquist, B.J., Soudzilovskaia, N.A., Ackerly, D. D., Anand, M., Atkin, O., Bahn, M., Baker, T.R., Baldocchi, D., Bekker, R., Blanco, C. C., Blonder, B., Bond, W.J., Bradstock, R., Bunker, D.E., Casanoves, F., Cavender-Bares, J., Chambers, J.Q., Chapin, F.S., Chave, J., Coomes, D., Cornwell, W.K., Craine, J.M., Dobrin, B.H., Duarte, L., Durka, W., Elser, J., Esser, G., Estiarte, M., Fagan, W.F., Fang, J., Fernández-Méndez, F., Fidelis, A., Finegan, B., Flores, O., Ford, H., Frank, D., Freschet, G.T., Fyllas, N.M., Gallagher, R.V., Green, W.A., Gutierrez, A.G., Hickler, T., Higgins, S.I., Hodgson, J.G., Jalili, A., Jansen, S., Joly, C. A., Kerkhoff, A.J., Kirkup, D., Kitajima, K., Kleyer, M., Klotz, S., Knops, J.M.H., Kramer, K., Kühn, I., Kurokawa, H., Laughlin, D., Lee, T.D., Leishman, M., Lens, F., Lenz, T., Lewis, S.L., Lloyd, J., Llusià, J., Louault, F., Ma, S., Mahecha, M.D., Manning, P., Massad, T., Medlyn, B.E., Messier, J., Moles, A.T., Müller, S.C., Nadrowski, K., Naeem, S., Niinemets, Ü., Nöllert, S., Nüske, A., Ogaya, R., Oleksyn, J., Onipchenko, V.G., Onoda, Y., Ordoñez, J., Overbeck, G., Ozinga, W.A., Patiño, S., Paula, S., Pausas, J.G., Peñuelas, J., Phillips, O.L., Pillar, V., Poorter, H., Poorter, L., Poschlod, P., Prinzing, A., Proulx, R., Rammig, A., Reinsch, S., Reu, B., Sack, L., Salgado-Negret, B., Sardans, J., Shiodera, S., Shipley, B., Siefert, A., Sosinski, E., Soussana, J.F., Swaine, E., Swenson, N., Thompson, K., Thornton, P., Waldram, M., Weiher, E., White, M., White, S., Wright, S.J., Yguel, B., Zaehle, S., Zanne, A.E., Wirth, C., 2011. TRY - a global database of plant traits. Global Change Biol. 17, 2905–2935. https://doi.org/10.1111/j.1365-2486.2011.02451.x.

Lee, H., Kim, S., 2016. Black-box classifier interpretation using decision tree and fuzzy logic-based classifier implementation. Int. J. Fuzzy Log. Int. Syst. 16, 27–35. https://doi.org/10.5391/IJFIS.2016.16.1.27.

Li, P., Dong, H., Zheng, C., Sun, M., Liu, A., Wang, G., Liu, S., Zhang, S., Chen, J., Li, Y., Pang, C., Zhao, X., 2017. Optimizing nitrogen application rate and plant density for improving cotton yield and nitrogen use efficiency in the North China Plain. PLOS ONE 12, e0185550.

Li, J., Xie, R.Z., Wang, K.R., Ming, B., Guo, Y.Q., Zhang, G.Q., Li, S.K., 2015. Variations in Maize Dry Matter, Harvest Index, and Grain Yield with Plant Density. Agronomy Journal 107, 829–834.

Liu, W., Hou, P., Liu, G., Yang, Y., Guo, X., Ming, B., Xie, R., Wang, K., Liu, Y., Li, S., 2020. Contribution of total dry matter and harvest index to maize grain yield—A multisource data analysis. Food and Energy Security 9, e256.

Lovász, L., Vempala, S., 2003. Hit-and-run Is Fast and Fun. preprint, Microsoft Research.

Ma, S., Churkina, G., Wieland, R., Gessler, A., 2011. Optimization and evaluation of the ANTHRO-BGC model for winter crops in Europe. Ecol. Model. 222, 3662–3679.

den Meersche, K.V., Soetaert, K., Oevelen, D.V., 2009. Xsample: an R function for sampling linear inverse problems. J. Stat. Software 30, 1–15. https://doi.org/10.18637/jss.v030.c01.

Moré, J.J., 1978. The Levenberg-Marquardt algorithm: implementation and theory. In: Numerical Analysis. Springer, pp. 105–116.

Nobel, A.B., 2002. Analysis of a complexity-based pruning scheme for classification trees. IEEE Trans. Inf. Theor. 48, 2362–2368.

Pásztor, L., Laborczi, A., Takács, K., Illés, G., Szabó, J., Szatmári, G., 2020. Progress in the elaboration of GSM conform DSM products and their functional utilization in Hungary. Geoderma Reg. 21, e00269 https://doi.org/10.1016/j.geodrs.2020.e00269.

Pericchi, L.R., Walley, P., 1991. Robust Bayesian credible intervals and prior ignorance. Int. Stat. Rev./Revue Int. de Stat. 1–23.

Pokovai, K., Fodor, N., 2019. Adjusting Ceptometer Data to Improve Leaf Area Index Measurements. Agronomy 9, 866.

Prescher, A.-K., Grünwald, T., Bernhofer, C., 2010. Land use regulates carbon budgets in eastern Germany: from NEE to NBP. Agric. For. Meteorol. 150, 1016–1025.

Prihodko, L., Denning, A.S., Hanan, N.P., Baker, I., Davis, K., 2008. Sensitivity, uncertainty and time dependence of parameters in a complex land surface model. Agric. For. Meteorol. 148, 268–287.

R Core Team, 2021. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.

Richardson, A.D., Williams, M., Hollinger, D.Y., Moore, D.J.P., Dail, D.B., Davidson, E.A., Scott, N.A., Evans, R.S., Hughes, H., 2010. Estimating parameters of a forest ecosystem C model with measurements of stocks and fluxes as joint constraints. Oecologia 164, 25–40. https://doi.org/10.1007/s00442-010-1628-y, 164, 25–40.

Running, S.W., Hunt, E.R.J., 1993. Generalization of a forest ecosystem process model for other biomes, BIOME-BGC, and an application for global-scale models. In: Ehleringer, J.R., Field, C. (Eds.), Scaling Physiological Processes: Leaf to Globe. Academic Press, San Diego, pp. 141–158.

Sadegh, M., Vrugt, J.A., 2013. Bridging the gap between GLUE and formal statistical approaches: approximate Bayesian computation. Hydrol. Earth Syst. Sci. 17, 4831–4850.

Sándor, R., Barcza, Z., Hidy, D., Lellei-Kovács, E., Ma, S., Bellocchi, G., 2016. Modelling of grassland fluxes in Europe: evaluation of two biogeochemical models. Agric. Ecosyst. Environ. 215, 1–19.

Sexton, J., Everingham, Y., Inman-Bamber, G., 2016. A theoretical and real world evaluation of two Bayesian techniques for the calibration of variety parameters in a sugarcane crop model. Environ. Model. Software 83, 126–142. https://doi.org/10.1016/j.envsoft.2016.05.014.

Stedinger, J.R., Vogel, R.M., Lee, S.U., Batchelder, R., 2008. Appraisal of the generalized likelihood uncertainty estimation (GLUE) method. Water Resour. Res. 44, W00B06. https://doi.org/10.1029/2008WR006822.

Stöckle, C.O., Nelson, R., 2013. Cropping Systems Simulation Model User's Manual, vol. 235.

Tarantola, A., 2005. Inverse Problem Theory and Methods for Model Parameter Estimation. SIAM.

Therneau, T., Atkinson, B., port, B.R. (producer of the Initial R., Maintainer 1999-2017), 2022. Rpart: Recursive Partitioning and Regression Trees.

Therond, O., Hengsdijk, H., Casellas, E., Wallach, D., Adam, M., Belhouchette, H., Oomen, R., Russell, G., Ewert, F., Bergez, J.-E., 2011. Using a cropping system model at regional scale: low-data approaches for crop management information and model calibration. Agric. Ecosyst. Environ. 142, 85–94.

Trudinger, C.M., Raupach, M.R., Rayner, P.J., Kattge, J., Liu, Q., Pak, B., Reichstein, M., Renzullo, L., Richardson, A.D., Roxburgh, S.H., Styles, J., Wang, Y.P., Briggs, P., Barrett, D., Nikolova, S., 2007. OptIC project: an intercomparison of optimization techniques for parameter estimation in terrestrial biogeochemical models. J. Geophys. Res. 112, G02027 https://doi.org/10.1029/2006jg000367.

Van Oijen, M., Rougier, J., Smith, R., 2005. Bayesian calibration of process-based forest models: bridging the gap between models and data. Tree Physiol. 25, 915–927.

Wallach, D., Palosuo, T., Thorburn, P., Hochman, Z., Gourdain, E., Andrianasolo, F., Asseng, S., Basso, B., Buis, S., Crout, N., Dibari, C., Dumont, B., Ferrise, R., Gaiser, T., Garcia, C., Gayler, S., Ghahramani, A., Hiremath, S., Hoek, S., Horan, H., Hoogenboom, G., Huang, M., Jabloun, M., Jansson, P.-E., Jing, Q., Justes, E., Kersebaum, K.C., Klosterhalfen, A., Launay, M., Lewan, E., Luo, Q., Maestrini, B., Mielenz, H., Moriondo, M., Nariman Zadeh, H., Padovan, G., Olesen, J.E., Poyda, A., Priesack, E., Pullens, J.W.M., Qian, B., Schütze, N., Shelia, V., Souissi, A., Specka, X., Srivastava, A.K., Stella, T., Streck, T., Trombi, G., Wallor, E., Wang, J., Weber, T.K. D., Weihermüller, L., de Wit, A., Wöhling, T., Xiao, L., Zhao, C., Zhu, Y., Seidel, S.J., 2021. The chaos in calibrating crop models: lessons learned from a multi-model calibration exercise. Environ. Model. Software 145, 105206. https://doi.org/10.1016/j.envsoft.2021.105206.

White, M.A., Thornton, P.E., Running, S.W., Nemani, R.R., 2000. Parameterization and sensitivity analysis of the BIOME-BGC terrestrial ecosystem model: net primary production controls. Earth Interact. 4, 1–85.

White, J.T., Knowling, M.J., Fienen, M.N., Siade, A., Rea, O., Martinez, G., 2022. A model-independent tool for evolutionary constrained multi-objective optimization under uncertainty. Environ. Environ. Model. Softw. 149, 105316 https://doi.org/10.1016/j.envsoft.2022.105316.

Wöhling, T., Gayler, S., Priesack, E., Ingwersen, J., Wizemann, H.-D., Högy, P., Cuntz, M., Attinger, S., Wulfmeyer, V., Streck, T., 2013. Multiresponse, multiobjective calibration as a diagnostic tool to compare accuracy and structural limitations of five coupled soil-plant models and CLM3. 5. Water Resour. Res. 49, 8200–8221.

Xiong, W., Holman, I., Conway, D., Lin, E., Li, Y., 2008. A crop model cross calibration for use in regional climate impacts studies. Ecol. Model. 213, 365–380.