Exploring Black-Box Parameter Calibration Techniques in System Dynamics Models

Tugba Uyar[a], Melis Tuğrul[b], Gönenç Yücel[a], Yaman Barlas[c], and Mustafa Gökçe Baydoğan[a]

[a]Department of Industrial Engineering, Boğaziçi University, Istanbul, Turkey
[b]Department of Industrial Engineering, Yıldız Technical University, Istanbul, Turkey
[c]Acıbadem University, Istanbul, Turkey

Abstract

Parameter calibration in system dynamics models can be challenging, particularly when models include nonlinear feedback structures. This study examines several black-box parameter calibration techniques in the context of system dynamics models. Two models are used in the experiments: a SEIR epidemiological model and a stock management model. Synthetic datasets are generated from the models and used as observed data for calibration. The calibration techniques are applied to estimate selected model parameters, and their performance is compared across the two different models. The results also highlight the importance of considering dynamic behavior in addition to numerical error measures when evaluating calibration outcomes. The applicability of the approach is further supported by a related research in which the same methods outperformed manual expert calibration of a larger model with limited real-world data.

Keywords: system dynamics; parameter calibration; black-box optimization; synthetic data; epidemiological model; inventory dynamics

1. Introduction

Parameter calibration is an important step in system dynamics modelling. It involves adjusting parameter values so that simulated output matches historical data as closely as possible

(Oliva, 2003). These parameters often include constants, initial conditions, and graphical table functions. Since exact values are rarely available for all parameters, they must be estimated by reducing the difference between simulated and observed values (Oliva, 2003). A well-calibrated model can increase confidence in the model structure and in the policy insights derived from it.

In earlier studies, calibration was often carried out manually through expert judgment and trial-and-error adjustments (Lyneis and Pugh III, 1996). The modeler would run the simulation, compare the output with reference data, and adjust parameter values to improve the fit. This approach may work for simple models, but it becomes difficult as model complexity increases (Lyneis and Pugh III, 1996). System dynamics models often involve nonlinear interactions and delays, meaning that a change in one parameter can affect others indirectly or with delays. In large parameter spaces, manual calibration may fail to identify well-fitting parameter sets and is often difficult to reproduce (Dangerfield and Roberts, 1996). The problem becomes more serious when the available data are limited or noisy, since this increases the role of subjective judgment (Vanni et al., 2011).

To overcome these limitations, researchers have adopted automated search and optimization methods for system dynamics model calibration (Eksin, 2009). In automated calibration, parameter estimation is treated as an optimization problem. An objective function, such as the sum of squared errors, is defined and then minimized (Dangerfield and Roberts, 1996). However, system dynamics models are not always suitable for gradient-based methods. They often include discontinuous functions, logical conditions, and graphical relationships such as table functions (Hearne, 1987). These features can produce non-smooth error surfaces with several local optima (Dangerfield and Roberts, 1996). Consequently, traditional nonlinear optimization routines can become highly inefficient in nontrivial and high-dimensional parameter spaces (Andrade and Duggan, 2020). Practitioners often attempt to overcome this difficulty by running the optimization algorithm from multiple starting points; however, as the number of parameters increases, so does the risk of exhausting computational resources before finding the optimal start (Andrade, 2021). Furthermore, gradient-based methods may require analytical information that is not available in standard simulation software (Dangerfield and Roberts, 1996).

For these reasons, researchers use derivative-free and black-box methods to calibrate system dynamics models (Parra et al., 2018; Dangerfield and Roberts, 1996). In these approaches, the optimizer does not require analytical derivatives. Instead, it repeatedly evaluates the simulation model and uses the resulting output to guide the search (Parra et al., 2018). In this setting, the model is treated as a black box (Rios and Sahinidis, 2013; Terayama and Tsuda, 2021). This input-output approach allows the optimizer to work without accessing or modifying the internal equations of the simulation model. This flexibility circumvents the limitations of classical optimization and enables the straightforward implementation of advanced algorithms—such as evolutionary strategies, Bayesian optimization, and Markov

chain Monte Carlo (MCMC) methods to explore complex parameter spaces more effectively (Andrade, 2021; Rahmandad et al., 2015).

Several black-box methods can be used for parameter estimation. Grid search evaluates predefined parameter combinations over a discretized search space. It is simple, but it becomes computationally expensive as the number of parameters increases. Random search samples parameter values from predefined probability distributions. Earlier work suggests that it can explore the parameter space more broadly than grid search in high-dimensional settings, especially when only a small number of parameters strongly affect the objective function (Bergstra and Bengio, 2012).

More advanced optimization methods include the Tree-structured Parzen Estimator (TPE), Gaussian Process (GP) optimization, and the Covariance Matrix Adaptation Evolution Strategy (CMA-ES). TPE is a Bayesian optimization method that models promising and less promising parameter regions based on previous evaluations. It then uses this information to select new candidates (Bergstra et al., 2011). The GP sampler is another Bayesian approach that constructs a probabilistic surrogate model of the objective function to sequentially guide the selection of new parameter configurations (Rasmussen and Williams, 2006). CMA-ES is an evolutionary algorithm that updates a multivariate normal search distribution over successive generations by adapting its mean and covariance matrix (Hansen and Ostermeier, 2001). It is well suited to non-separable problems because it accounts for correlations between variables without requiring gradient information (Hansen and Ostermeier, 2001).

This study aims to replace manual calibration with a more systematic algorithmic search process. The model is treated as a black box, and the analysis is based on its input-output behavior. Five methods are compared: Grid Search, Random Search, TPE, GP, and CMA-ES. These methods are applied to two generic system dynamics models using synthetic datasets generated under controlled conditions. A more detailed presentation of the generic models, experimental design, and additional results can be found in Uyar (2025).

The methods are compared in terms of calibration accuracy, computational cost, and consistency. In addition, the results draw attention to the importance of assessing whether calibrated models reproduce the underlying dynamic behavior of the system, beyond achieving low numerical error values. The study aims to provide system dynamics modelers with a comparative analysis of automated calibration methods within an input-output framework. The applicability of these methods to a larger and more complex model calibrated with limited real-world data is also discussed based on findings from a related study (Tugrul, 2025).

1.1. SEIR Model

The first model used in the experiments is a Susceptible–Exposed–Infectious–Recovered (SEIR) epidemiological model adapted from the COVID-19 system dynamics model developed by (Struben, 2020). In addition to the standard SEIR structure, the model also includes a deceased population to represent deaths caused by the disease. The population is therefore divided into susceptible, exposed, infectious, recovered, and deceased compartments.

In this formulation, individuals in the exposed compartment are also assumed to contribute to disease transmission, unlike in the classical SEIR structure. This assumption reflects COVID-19 modeling approaches that allow partial infectiousness during the exposed stage (Leontitsis et al., 2021).

The infection process introduces nonlinear feedback, as the number of infectious and exposed individuals influences the rate at which susceptible individuals become exposed. As a result, the model produces typical epidemic dynamics characterized by growth, peak, and decline in the number of infections.

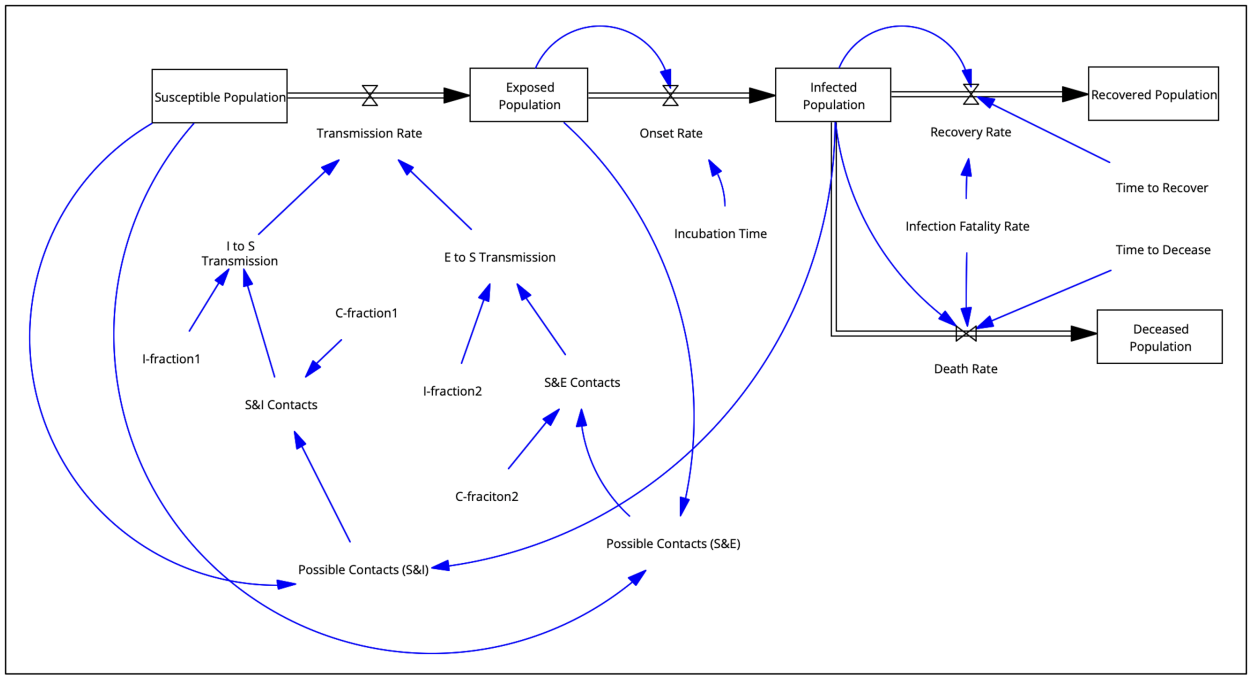


Figure 1. SEIR model structure.

1.2. Stock Management Model

The second model used in the experiments is a stock management model representing inventory control dynamics. The model includes an inventory stock that is adjusted through

production decisions based on expected demand and desired inventory levels.

The model contains feedback loops and adjustment delays between inventory levels and production rates. These delays may generate oscillatory behavior in the inventory level over time, which makes the model suitable for testing calibration techniques under different dynamic conditions.

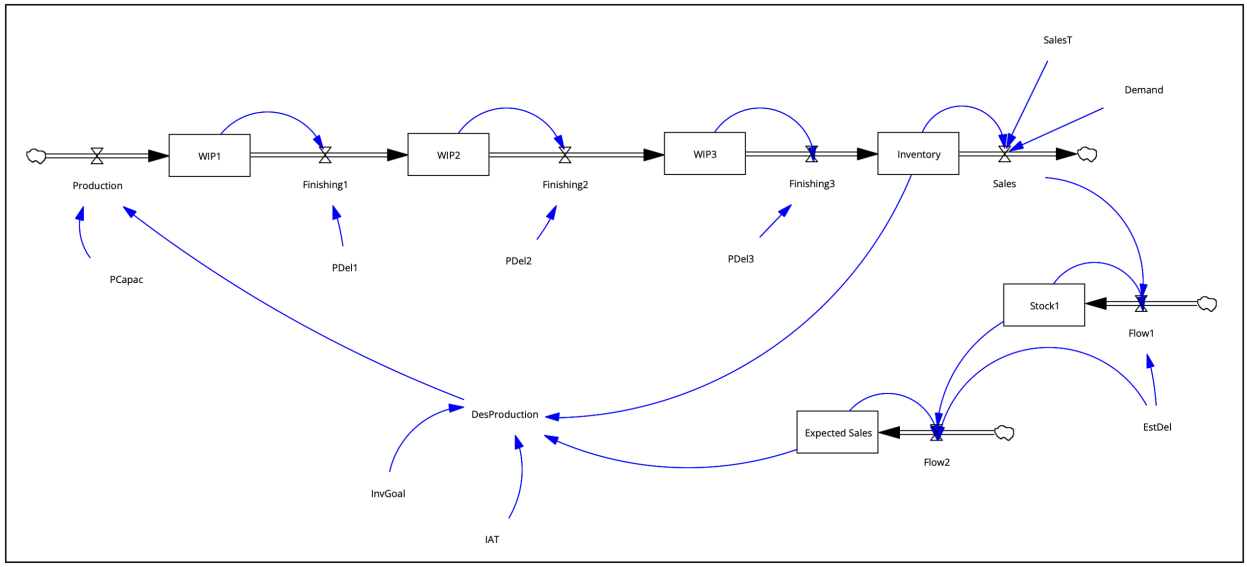


Figure 2. Stock management model structure.

2. Calibration Techniques

Several black-box parameter calibration techniques are evaluated in this study. These techniques differ in how they explore the parameter space and search for parameter values that reproduce the observed behavior of the models. The following approaches are considered: Grid Search, Random Search, Tree-structured Parzen Estimator (TPE), Gaussian Process (GP) Sampler, and Covariance Matrix Adaptation Evolution Strategy (CMA-ES).

2.1. Grid Search

Grid Search explores the parameter space by evaluating combinations of parameter values defined on a predefined grid. Each parameter is assigned a discrete set of possible values, and the model is simulated for every combination of these values. Although this approach provides systematic coverage of the search space, it becomes computationally expensive as the number of parameters increases.The algorithmic process is defined as follows: If the search space Λ is indexed by K configuration variables, the analyst chooses a discrete set of

values $L^{(k)}$ for each variable $k = 1, \ldots, K$. The set of all experimental trials is formed by calculating the Cartesian product of these sets. The total number of trials S evaluated by the algorithm is given by:

$$S = \prod_{k=1}^{K} |L^{(k)}| \tag{1}$$

For each configuration in the grid, the objective function $\Psi(\lambda)$ is evaluated, and the combination yielding the lowest error is selected as $\hat{\lambda}$ (Bergstra and Bengio, 2012). While Grid Search is simple to implement and easily parallelizable, the exponential growth of S relative to K makes the algorithm suffer severely from the curse of dimensionality (Bergstra and Bengio, 2012).

2.2. Random Search

Random Search is a non-adaptive stochastic strategy where trials are chosen as independent draws from a specified probability density over the configuration space Λ (Bergstra and Bengio, 2012). Instead of evaluating combinations on a fixed grid, parameter vectors $\lambda^{(i)}$ are generated randomly for $i = 1, \ldots, S$. The objective function $\Psi(\lambda^{(i)})$ is evaluated for each independent sample, and the one that minimizes the objective is returned.

Mathematical and empirical analyses indicate that Random Search is more efficient than Grid Search for high-dimensional problems because objective functions often have a low effective dimensionality (Bergstra and Bengio, 2012). If an objective function is highly sensitive to only a small subset of parameters, uniformly random draws provide a much denser coverage of those critical dimensions compared to a discretized grid, preventing wasted trials on unimportant dimensions (Bergstra and Bengio, 2012).

2.3. Tree-structured Parzen Estimator (TPE)

The Tree-structured Parzen Estimator (TPE) is a sequential model-based optimization algorithm. While standard surrogate methods model the conditional probability $p(y|x)$ where x represents the parameters and y is the objective value, TPE models the distributions $p(x|y)$ and $p(y)$ using non-parametric density estimation (Bergstra et al., 2011).

The TPE algorithm defines $p(x|y)$ by splitting the observations into two distinct densities based on a threshold y^*:

$$p(x|y) = \begin{cases} \ell(x) & \text{if } y < y^* \\ g(x) & \text{if } y \geq y^* \end{cases} \tag{2}$$

Here, y^* is chosen to be a specific quantile γ of the observed objective values, such that $p(y < y^*) = \gamma$. The density $\ell(x)$ is formed using the parameter configurations that yielded a

loss lower than y^*, whereas $g(x)$ is formed using the remaining observations (Bergstra et al., 2011).

To select the next candidate x^*, the TPE algorithm maximizes the Expected Improvement (EI). Based on the parameterization $p(x|y)p(y)$, maximizing the EI is mathematically equivalent to finding points with high probability under $\ell(x)$ and low probability under $g(x)$:

$$EI_{y^*}(x) \propto \left(\gamma + \frac{g(x)}{\ell(x)}(1-\gamma)\right)^{-1} \tag{3}$$

On each iteration, the algorithm draws multiple candidates according to $\ell(x)$, evaluates them according to the ratio $g(x)/\ell(x)$, and returns the candidate that yields the greatest expected improvement (Bergstra et al., 2011).

2.4. Gaussian Process (GP) Sampler

The Gaussian Process (GP) sampler is a model-based Bayesian optimization technique. It constructs a probabilistic surrogate model of the objective function based on previously evaluated parameter sets (Rasmussen and Williams, 2006). A GP is formally defined as a collection of random variables, any finite number of which have a joint Gaussian distribution. It is completely specified by its mean function $\mu(x)$ and covariance function $k(x, x')$:

$$f(x) \sim \mathcal{GP}(\mu(x), k(x, x')) \tag{4}$$

Assuming access to noisy function values $y = f(x) + \epsilon$ with an independent identically distributed Gaussian noise variance λ^2, the predictive posterior equations for the mean $\mu(x)$ and variance $\sigma^2(x)$ at a new unexplored point x, given the observations $X_{1:n}$ and y, are updated analytically:

$$\mu(x) = k(x, X_{1:n})[K(X_{1:n}, X_{1:n}) + \lambda^2 I]^{-1} y \tag{5}$$

$$\sigma^2(x) = k(x, x) - k(x, X_{1:n})[K(X_{1:n}, X_{1:n}) + \lambda^2 I]^{-1} k(X_{1:n}, x) \tag{6}$$

where K is the covariance matrix of the evaluated inputs (Rasmussen and Williams, 2006). To guide the selection of new parameter candidates, the algorithm maximizes an acquisition function, such as the Expected Improvement (EI), which balances the exploitation of regions with low predicted error and the exploration of areas with high uncertainty (Rasmussen and Williams, 2006).

2.5. Covariance Matrix Adaptation Evolution Strategy (CMA-ES)

CMA-ES is an evolutionary optimization algorithm designed for continuous parameter spaces. In this method, candidate solutions are generated by sampling from a multivariate normal distribution. The mean and covariance matrix of this distribution are iteratively updated based on the performance of previously evaluated solutions, allowing the search distribution to adapt to the structure of the objective function (Hansen, 2006). At each generation g, a population of λ new search points is generated according to:

$$x_k^{(g+1)} \sim m^{(g)} + \sigma^{(g)} \mathcal{N}(0, C^{(g)}) \quad \text{for } k = 1, \ldots, \lambda \tag{7}$$

where $m^{(g)} \in \mathbb{R}^n$ is the mean value of the search distribution, $\sigma^{(g)} \in \mathbb{R}_+$ is the overall step size, and $C^{(g)} \in \mathbb{R}^{n \times n}$ is the symmetric and positive definite covariance matrix (Hansen, 2006).

After evaluating the objective function for all $x_k^{(g+1)}$, the algorithm selects the best μ individuals. The new mean $m^{(g+1)}$ is calculated as a weighted average of these successful search points:

$$m^{(g+1)} = \sum_{i=1}^{\mu} w_i x_{i:\lambda}^{(g+1)} \tag{8}$$

where $w_i > 0$ are the recombination weights satisfying $\sum_{i=1}^{\mu} w_i = 1$, and $x_{i:\lambda}^{(g+1)}$ denotes the i-th best individual (Hansen, 2006).

3. Experimental Design

3.1. Synthetic Data Creation

Synthetic datasets are generated from the simulation outputs of the models and used as "observed data" in the calibration experiments. Using synthetic data allows the true parameter values of the models to be known, which makes it possible to evaluate how well the calibration techniques recover these values.

The models are first simulated using predefined parameter values, and the resulting time series are treated as the baseline model behavior. "Observed datasets" are then generated by adding noise to the simulated outputs. This approach allows controlled experimentation while preserving the underlying dynamics of the original models.

The generated datasets are used as the target observations during the calibration process. Calibration techniques aim to identify parameter values that reproduce the observed data. The estimated parameters are then compared with the true parameter values used to generate

the synthetic data.

3.2. Noise Generation

To simulate imperfections commonly observed in real-world data, noise is added to the synthetic datasets generated from the model simulations. The noise affects only the observed data used during calibration and does not alter the underlying dynamics of the models.

Two types of noise are considered in the experiments: pure noise and autocorrelated noise. Pure noise represents independent random fluctuations in observations, while autocorrelated noise introduces temporal dependence between consecutive observations.

Pure noise is generated by adding random disturbances independently to each observation. In contrast, autocorrelated noise introduces correlation between successive time steps, reflecting situations where measurement errors or reporting delays persist over time.

By incorporating these two noise structures, the calibration experiments evaluate how different techniques perform under varying levels of data imperfection.

Examples of the generated noisy observed datasets are shown in Figures 3 and 4. In these figures, the original model outputs are plotted together with the corresponding noisy observations. The SEIR model exhibits smoother deviations due to autocorrelated noise, whereas the stock management model shows more irregular fluctuations under pure noise.

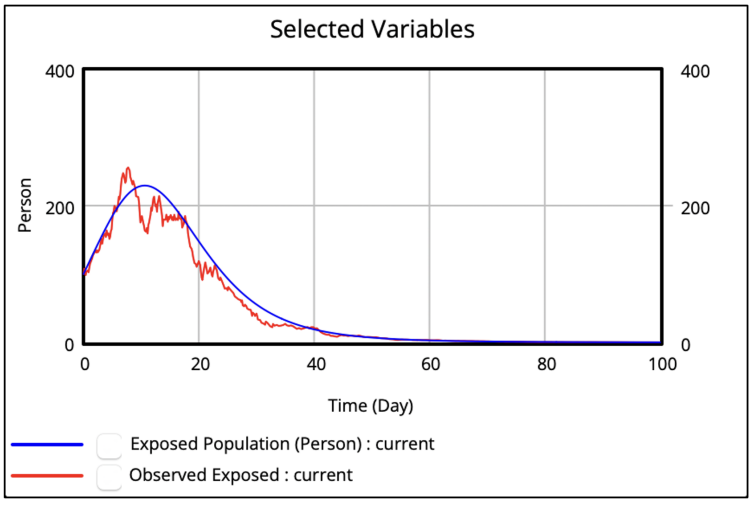


Figure 3. Synthetic observed data generated by adding strong autocorrelated noise to the SEIR model output.

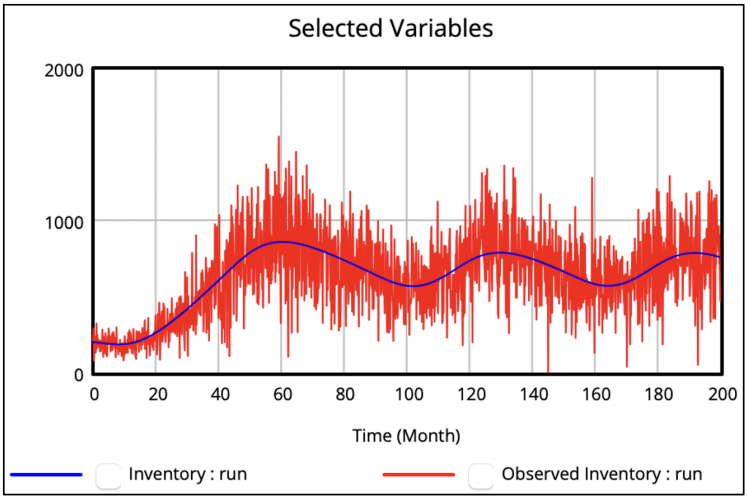


Figure 4. Synthetic observed data generated by adding strong pure noise to the stock management model output.

3.3. Calibration Parameters

In the calibration experiments, selected parameters of the models are treated as unknown and estimated using the calibration techniques described earlier. The remaining parameters are kept fixed at their baseline values.

For the SEIR model, calibration focuses on parameters that directly influence disease transmission and progression dynamics. These include the *Infection Fatality Rate*, *Incubation Time*, and *Time to Recover*. These parameters play a central role in shaping the epidemic dynamics and therefore represent meaningful targets for calibration.

For the stock management model, calibration focuses on the production delay parameters *PDel1*, *PDel2*, and *PDel3*. These parameters represent delays in the production process and influence how quickly the production system responds to inventory imbalances.

Parameter ranges are defined based on plausible values derived from the model structure and prior studies. During calibration, the algorithms search within these predefined ranges to identify parameter values that reproduce the observed model behavior. Table 1 summarizes the parameter values and ranges used in the calibration experiments.

For consistency across methods, the total number of evaluations is kept the same for all calibration techniques. Grid Search evaluates all possible combinations of parameter values, resulting in 125 evaluations. The other calibration techniques are also configured to perform 125 evaluations to ensure a fair comparison.

Table 1. Parameter values and ranges used in the calibration experiments

Model	Parameter	Grid Search	Other Algorithms
SEIR	Infection Fatality Rate	[0.01, 0.05, 0.1, 0.15, 0.2]	0.01 – 0.20
SEIR	Incubation Time	[1, 5, 10, 15, 20]	1 – 20 days
SEIR	Time to Recover	[1, 7, 14, 21, 25]	1 – 25 days
Stock management	PDel1	[0.5, 5, 10, 15, 20]	(0.5, 20)
Stock management	PDel2	[0.5, 5, 10, 15, 20]	(0.5, 20)
Stock management	PDel3	[0.5, 5, 10, 15, 20]	(0.5, 20)

3.4. Evaluation Metrics

The performance of the calibration techniques is evaluated using both quantitative and qualitative criteria. First, the mean squared error (MSE) between simulated and observed data is used to assess how closely the model output matches the observed values, providing a standard measure of numerical fit. Second, the distance between the estimated parameter vector and the true parameter vector is calculated. Since the experiments are based on synthetic data generated from known parameter values, this metric allows direct evaluation of parameter recovery performance.

The resulting model behavior is also examined by comparing the simulated outputs produced by the calibrated parameter sets with the reference model behavior. This evaluation focuses on key characteristics such as growth, peak, decline, and oscillatory structures, depending on the model. To assess computational performance, execution times of the calibration techniques are recorded, and the number of trials required to reach the best solution within the predefined search budget is analyzed. Since each method is run with a fixed number of trials, this measure provides insight into how quickly different techniques converge to high-quality solutions. Calibration outcomes are also evaluated under different noise structures and data availability conditions. In particular, an extreme condition test is conducted using a limited number of observations to examine how the algorithms behave when the available data is insufficient to clearly reveal the underlying dynamics.

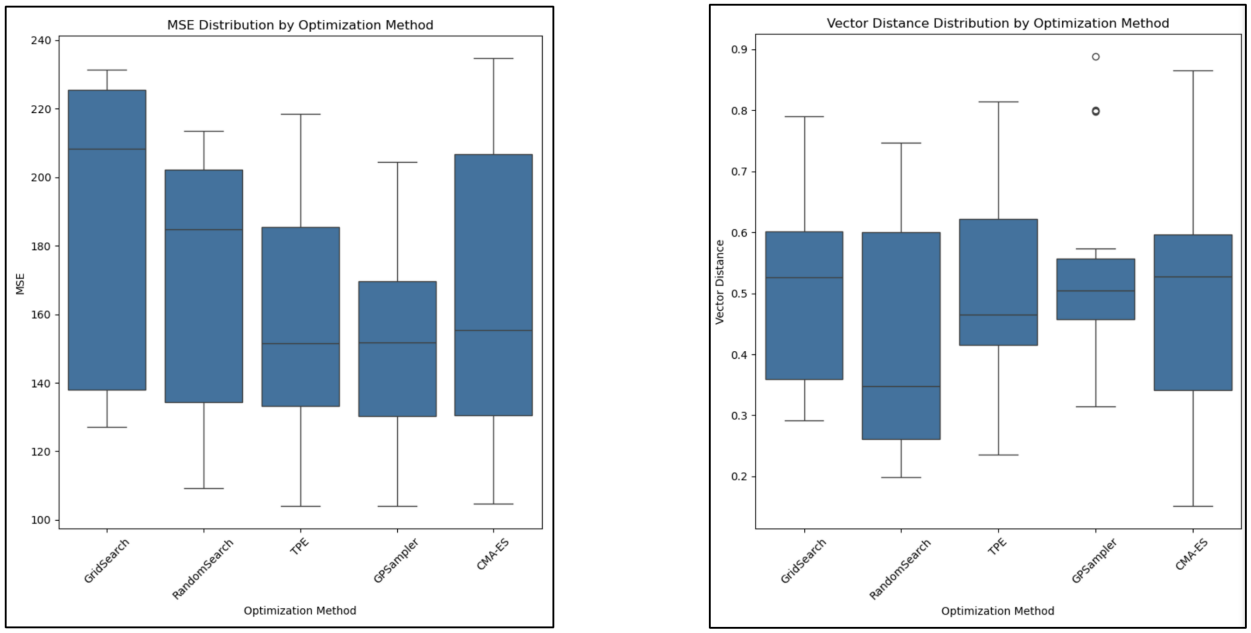
These criteria allow the calibration performance to be evaluated from multiple perspectives, including numerical accuracy, parameter recovery, dynamic behavior, and computational efficiency.

4. Results

This section presents representative results obtained for two system dynamics models under five experimental settings: pure noise, strong pure noise, mild autocorrelated noise, strong

autocorrelated noise, and strong pure noise with sparse data. For each model and each experimental setting, five different SD seed values are used to generate five distinct observed synthetic datasets. In addition, five different BBO seed values are used for each calibration technique in order to vary the search process while keeping the comparison across methods consistent. Grid Search, being deterministic, is not affected by BBO seed values. Therefore, for a given model and experimental setting, it produces five distinct best parameter sets corresponding to the five observed datasets generated with different SD seed values. In contrast, the stochastic calibration techniques are affected by both SD seed values and BBO seed values, yielding 25 best parameter sets for each model-setting combination.

Rather than averaging these outcomes into a single value, all results are retained and examined in terms of their distribution and consistency. For this reason, the analysis is based primarily on box-plots and representative dynamic behavior figures, which together provide a clearer view of both numerical performance and variability across runs. Figure 5 presents the distribution of calibration outcomes for the SEIR model under the strong autocorrelated noise case. Each box plot represents the results obtained from multiple runs using different SD seed and BBO seed combinations. The figure demonstrates that calibration performance varies not only across methods but also across runs of the same method.



(a) Mean Squared Error (MSE)

(b) Vector Distance

Figure 5. Distribution of calibration outcomes for the SEIR model under strong autocorrelated noise.

Table 2 provides a complementary perspective by presenting execution times for a representative experimental case. While execution times vary across different cases, a consistent

pattern is observed in which most methods exhibit comparable computational costs, whereas the GP sampler requires substantially longer execution times.

Table 2. Execution time comparison across optimization methods for the stock management model under strong pure noise.

Method	**Execution Time**
Grid Search	1 minute 59 seconds
Random Search	1 minute 55 seconds
TPE	2 minutes 6 seconds
GP Sampler	11 minutes 8 seconds
CMA-ES	2 minutes

A comprehensive set of results covering all experimental settings is provided in Uyar (2025). These results, based on the SEIR and stock management models examined in this study, suggest that black-box optimization techniques are suitable for calibrating medium-scale system dynamics models under different noise structures and data conditions.

However, numerical performance alone is not sufficient to fully assess calibration quality. In system dynamics models, it is essential to evaluate whether the calibrated parameters reproduce the characteristic behavior of the system. Therefore, the following analysis focuses on the dynamic patterns generated by the calibrated models. Based on the distribution of results obtained across multiple runs, the parameter sets associated with the lowest and highest MSE values are selected to illustrate representative calibration outcomes.

Overall, the calibration techniques are generally able to approximate the observed synthetic data generated from the models. However, the results also indicate that a low error value does not necessarily guarantee that the dynamic behavior of the model is correctly reproduced. In some cases, parameter sets that produce relatively low MSE values still generate noticeably different behaviors.

For the SEIR model, all calibration techniques are able to capture the general epidemic pattern characterized by growth, peak, and decline in the number of infections. However, differences between the simulated and observed trajectories can still occur, particularly in the magnitude of the epidemic peak. In Figure 6, the dashed blue line represents the reference trajectory generated using the true parameter values, while the model is calibrated using noisy observed data (shown as red markers in Figure 3). Therefore, a successful calibration is expected to reproduce the dynamic behavior of the dashed blue line. As shown in Figure 6, the trajectories correspond to the lowest MSE values obtained among the 25 calibration outcomes. None of the methods fully recover the peak magnitude associated with the true parameter values, which is likely due to the distortion introduced by noise in the observed data.

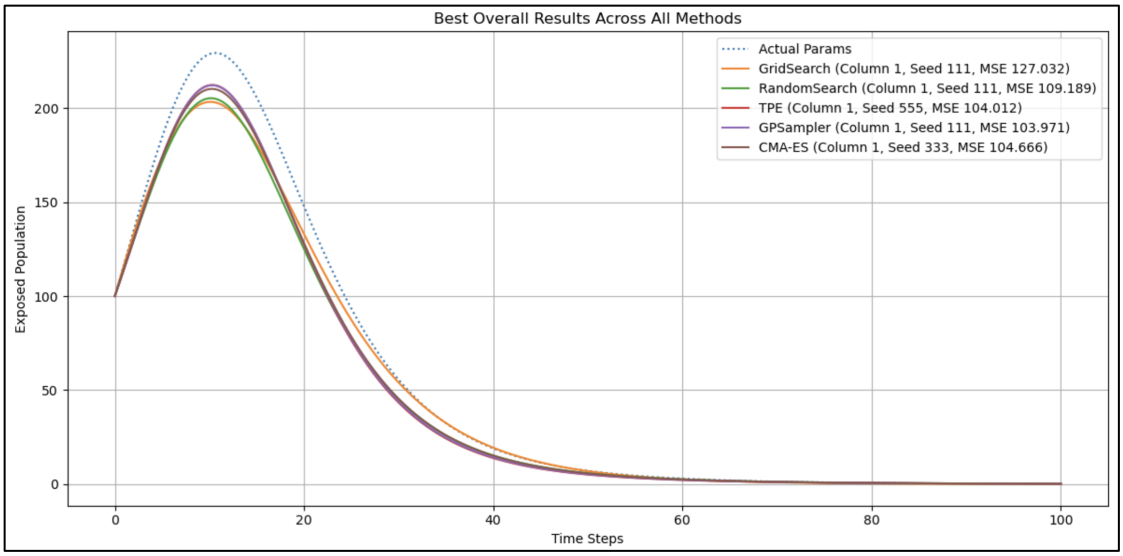


Figure 6. Dynamic behaviors corresponding to the lowest MSE among 25 calibration outcomes (SEIR model, strong autocorrelated noise).

For the stock management model, the results are evaluated based on how well the oscillatory behavior of the system is reproduced. Although several calibration techniques capture the general oscillatory pattern, noticeable deviations from the original trajectory can occur. In Figure 7, the dashed blue line represents the reference trajectory generated using the true parameter values, while the calibration is performed using noisy observed data (shown as red markers in Figure 4). As illustrated in Figure 7, differences in both the amplitude and phase of the oscillations become more visible when calibration outcomes with higher MSE values are considered.

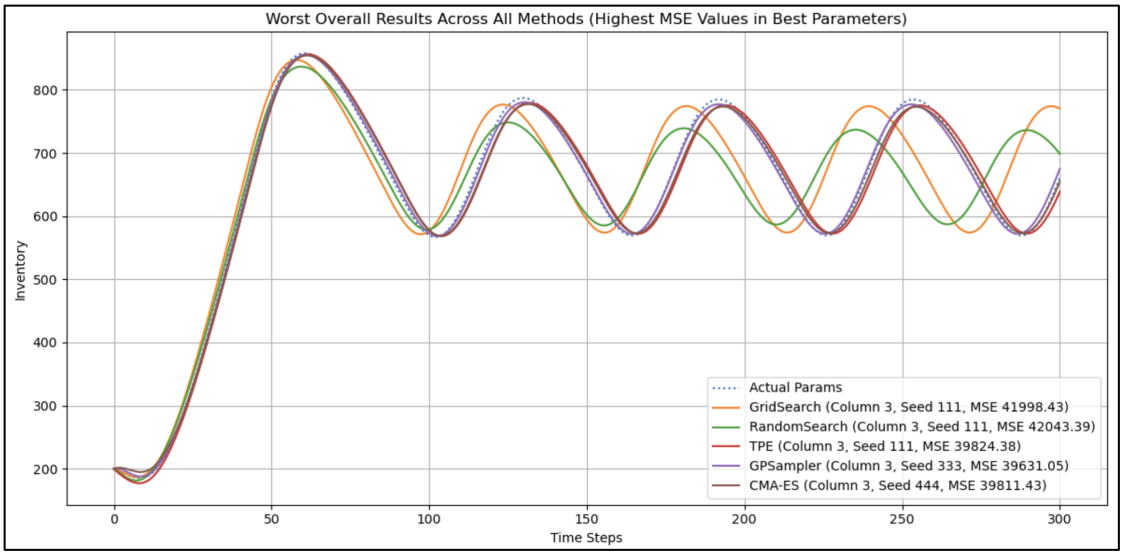
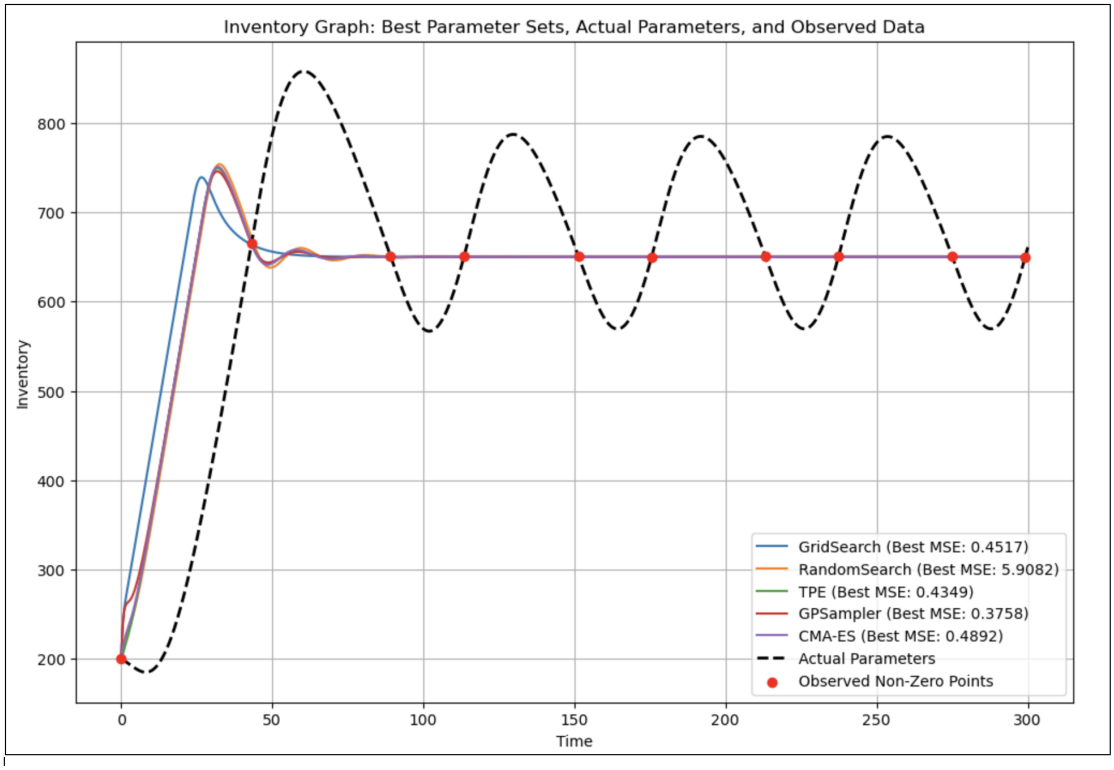


Figure 7. Dynamic behaviors corresponding to the highest MSE among 25 calibration outcomes (stock management model, strong pure noise).

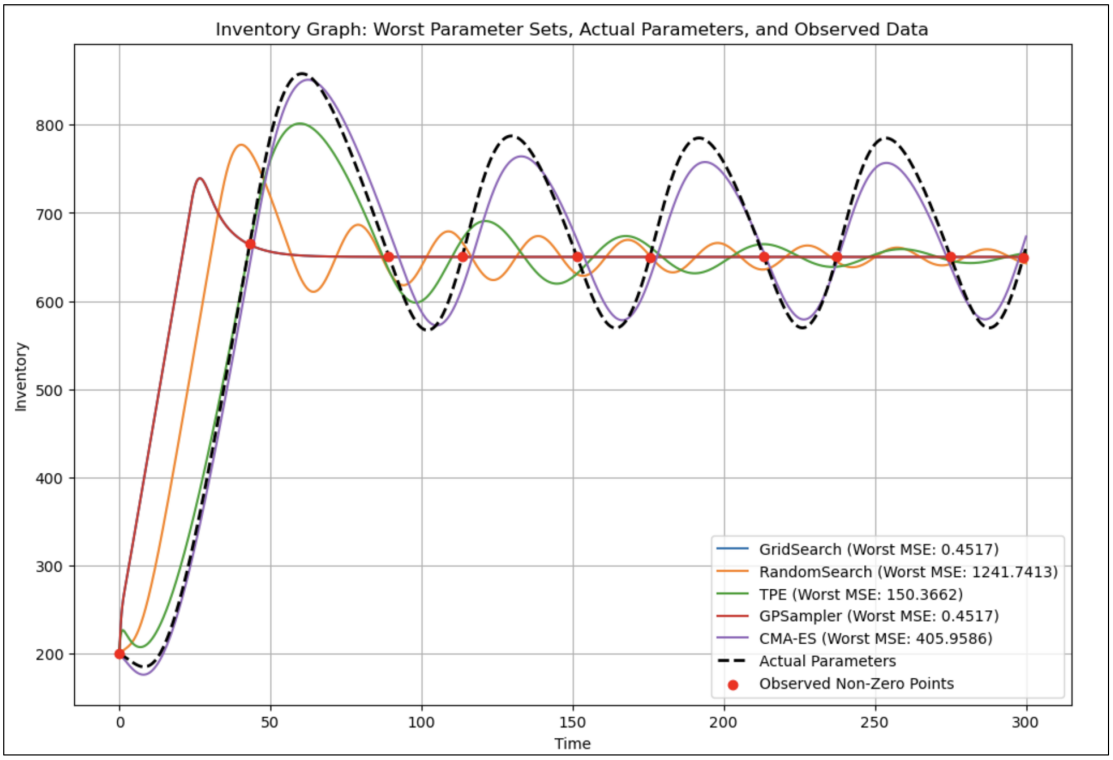
To further examine the calibration outcomes for the stock management model, an extreme condition test is also included. This test focuses on the *sparse* data setting, where the number of observed data points is substantially reduced. The purpose is to evaluate how the calibrated parameter sets behave when calibration is performed under a more restrictive data condition.

To illustrate this effect, the parameter sets and dynamics associated with the lowest and highest MSE values across the calibration outcomes are presented in Figure 8. In this extreme condition scenario, only a limited number of noisy observed data points are available, and the figure presents these observations directly, without including the reference trajectory generated from the true parameter values. The calibration algorithms are therefore fitted to these sparse observations.

Although the lowest MSE values indicate a good numerical fit, the resulting trajectories do not always reproduce the expected oscillatory behavior of the stock management model. This result highlights a key limitation of relying solely on statistical error metrics: a parameter set that minimizes MSE may still fail to capture the true dynamic behavior of the system.



(a) Best mean squared error (MSE)



(b) Worst mean squared error (MSE)

Figure 8. Best and worst MSE results of the parameter sets obtained in the extreme condition test for the stock management model.

5. Further Work and Discussion

The calibration approach used in this study was also applied to evaluate the capability of black-box algorithms in calibrating a larger system dynamics model with limited real-world data in a related research (Tugrul, 2025). The SD model comprises four interacting subsystems, each representing a distinct risk factor, with the population disaggregated by age group and sex. In each subsystem, seven parameters were calibrated simultaneously against six demographic output series, resulting in a substantially higher-dimensional optimization task than the settings examined here. We are unable to provide more information about this SD model at this point, due to confidentiality issues. The model had previously been calibrated manually by its developers through expert judgment and iterative adjustment, and this manually calibrated parameter set served as a baseline for comparison. Despite the limited availability of real-world data, all four black-box optimization methods (Grid Search, Random Search, TPE and CMA-ES) produced lower error values than this manually calibrated baseline, supporting the applicability of the approach beyond the controlled settings examined in the present study. Consistent with the findings reported here, however, achieving a low error value did not always guarantee that the calibrated model correctly reproduced the expected dynamic behavior, suggesting that visual inspection of model trajectories remains necessary alongside numerical error metrics. A similar pattern was also observed with respect to noise: as noise intensity increased, the performance differences among calibration methods diminished, which aligns with the findings of the present study.

The calibration experiments show that the structural properties of system dynamics models influence calibration outcomes. Unlike purely statistical models, system dynamics models often contain nonlinear feedback structures that generate complex temporal patterns such as epidemic waves or oscillatory inventory cycles. These structures can make calibration sensitive to small parameter variations, particularly in models with strong feedback loops.

This sensitivity becomes more visible in models with oscillatory dynamics. In such systems, small differences in parameter values may lead to noticeable changes in the timing, amplitude, or damping of oscillations. As a result, calibration procedures that focus primarily on reducing numerical error may still produce behaviors that differ in their qualitative dynamic characteristics.

Calibration outcomes also depend on how much data is available and how it is structured. When the number of observations is limited, different parameter sets may produce model behaviors that appear statistically similar within the available data range while diverging outside that range. This situation makes it more difficult to identify parameter sets that reproduce the full dynamic behavior of the model. The introduction of noise further complicates this challenge, as it can flatten the error surface and reduce the distinction between well-fitting and poorly-fitting parameter sets.

6. Conclusion

This study evaluated several black-box calibration techniques for parameter estimation in system dynamics models using synthetic datasets. Two models with different dynamic characteristics were used to examine how the algorithms perform under varying experimental conditions, including noisy observations and limited data availability.

The results indicate that black-box optimization techniques can successfully calibrate medium-scale system dynamics models, although their performance varies across models and experimental settings. More importantly, the experiments show that while many calibration techniques are capable of achieving low numerical error values, this does not always imply that the correct system dynamics are reproduced. In particular, models with oscillatory behavior may exhibit significantly different dynamics even when calibration metrics suggest a good fit.

In a related research in which we applied a subset of these methods to a large-scale system dynamics model with real-world data produced broadly consistent findings with those of the present study, as discussed in Section 5. In that setting, all four automated methods (Grid Search, Random Search, TPE and CMA-ES) achieved lower error values than the manually calibrated baseline despite sparse observational data. The higher-dimensional parameter space in that application also highlighted the importance of behavioral assessment alongside numerical error measures.

These findings emphasize the importance of evaluating calibration outcomes not only through statistical error measures but also through the dynamic behavior generated by the model. Potential directions for future research include pattern-oriented validation, hierarchical calibration schemes, and hybrid optimization approaches.

References

Andrade, J. (2021). A bayesian approach to calibrate system dynamics models using hamiltonian monte carlo. *System Dynamics Review*, 37.

Andrade, J. and Duggan, J. (2020). An evaluation of hamiltonian monte carlo performance to calibrate age-structured compartmental seir models to incidence data. *Epidemics*, 33:100415.

Bergstra, J., Bardenet, R., Bengio, Y., and Kégl, B. (2011). Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2546–2554.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(1):281–305.

Dangerfield, B. and Roberts, C. (1996). An overview of strategy and tactics in system dynamics optimization. *The Journal of the Operational Research Society*, 47(3):405–423.

Eksin, C. (2009). Application of genetic algorithms to analysis and policy design in system dynamics. Master's thesis, Boğaziçi University.

Hansen, N. (2006). The cma evolution strategy: A comparing review. In Lozano, J., Larrañaga, P., Inza, I., and Bengoetxea, E., editors, *Towards a New Evolutionary Computation*, pages 75–102. Springer, Berlin.

Hansen, N. and Ostermeier, A. (2001). Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195.

Hearne, J. W. (1987). An approach to resolving the parameter sensitivity problem in system dynamics methodology. *Applied Mathematical Modelling*, 11:315–318.

Leontitsis, A., Senok, A., Alsheikh-Ali, A., Al Nasser, Y., Loney, T., and Alshamsi, A. (2021). Seahir: A specialized compartmental model for covid-19. *International Journal of Environmental Research and Public Health*, 18(5):2667.

Lyneis, J. M. and Pugh III, A. L. (1996). Automated vs "hand" calibration of system dynamics models: An experiment with a simple project model. In *Proceedings of the 1996 International System Dynamics Conference*, volume 1, pages 321–324.

Oliva, R. (2003). Model calibration as a testing strategy for system dynamics models. *European Journal of Operational Research*, 151(3):552–568.

Parra, J. F., Jaramillo, P., and Arango-Aramburo, S. (2018). Metaheuristic optimization methods for calibration of system dynamics models. *Journal of Simulation*, 12(2):190–209.

Rahmandad, H., Oliva, R., and Osgood, N., editors (2015). *Analytical Methods for Dynamic Modelers*. The MIT Press, Cambridge, Massachusetts.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. MIT Press, Cambridge.

Rios, L. M. and Sahinidis, N. V. (2013). Derivative-free optimization: a review of algorithms and comparison of software implementations. *Journal of Global Optimization*, 56(3):1247–1293.

Struben, J. (2020). The coronavirus disease (covid-19) pandemic: Simulation-based assessment of outbreak responses and postpeak strategies. *System Dynamics Review*, 36(3):247–293.

Terayama, K. and Tsuda, K. (2021). Black-box optimization for automated discovery. *Accounts of Chemical Research*.

Tugrul, R. M. (2025). A comparative study of black-box optimization techniques in system dynamics parameter estimation with limited real data. Master's thesis, Boğaziçi University, Istanbul, Turkey.

Uyar, T. (2025). Exploring black-box parameter calibration techniques on generic system dynamics models, using synthetic data. Master's thesis, Bogazici University, Istanbul, Turkey.

Vanni, T., Karnon, J., Madan, J., White, R. G., Edmunds, W. J., Foss, A. M., and Legood, R. (2011). Calibrating models in economic evaluation. *PharmacoEconomics*, 29(1):35–49.