

Assessing the Performance of Various Stochastic Optimization Methods on Chemical Kinetic Modeling of Combustion

Chowdhury Ashraf and Jim Pfaendtner*

Cite This: *Ind. Eng. Chem. Res.* 2020, 59, 19212–19225

Read Online

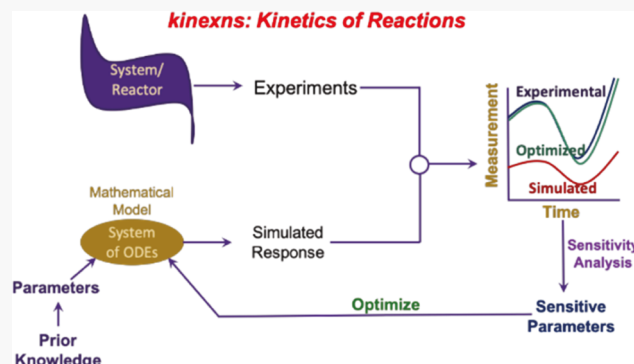
ACCESS |

Metrics & More

Article Recommendations

Supporting Information

ABSTRACT: The solution to chemical kinetic models for a particular reactor configuration is usually composed of a set of stiff ordinary differential equations (ODEs), where a number of rate parameters have to be estimated to fit with experimental observations. This presents a twofold challenge—first, to solve the stiff set of ODEs accurately and efficiently, and second, to estimate the model parameters precisely by optimizing the objective function. In recent years, stochastic optimization methods for parameter estimation have gained popularity over the classical optimization methods as the former do not require a reasonable initial guess and have the capability to escape local minima. In this study, we systematically examined 10 different stochastic optimization algorithms and evaluated their performance to estimate the model parameters for the previously developed propane oxidation mechanism, popularly known as the San Diego mechanism. In doing this, we developed an open source python package kinexns to efficiently solve the kinetic model using CVode solver, perform sensitivity analysis to determine important model parameters, and optimize the model parameters by using the different stochastic methods. The different algorithms we considered are Monte Carlo (MC), Latin hypercube sampling (LHS), maximum likelihood estimation (MLE), Markov chain Monte Carlo (MCMC), shuffled complex evolution algorithm (SCE-UA), simulated annealing (SA), robust parameter estimation (ROPE), artificial bee colony (ABC), fitness scaled chaotic artificial bee colony (FSCABC), and dynamically dimensioned search algorithm (DDS). The results indicated that the MLE and DDS provide more reliable parameter approximation among all of the algorithms evaluated.



1. INTRODUCTION

Mechanistic kinetic models are essential tools in understanding chemical and biochemical processes. They usually consist of reaction mechanisms with numerous elementary reactions whose rate parameters are often unknown or taken from the existing literature. In some cases, these parameters are estimated through quantum chemistry calculations using transition state or more advanced kinetic rate theories. However, regardless of how the rate parameters are obtained, when combined together, they seldom realistically fully predict experimental results as the operating condition, component molecular composition, and uncertainty in rate parameters may vary significantly.^{1,2} The uncertainty in rate parameters derives from variation in experimental measurements, increasing the difficulty of creating and training reliable and transferable kinetic models. For example, the rate constant of the reaction $H + O_2 \leftrightarrow O + OH$ has been examined at least 77 times in the literature.³ Nevertheless, mechanistic models are preferred over empirical models (often called “lumped” models) since the parameters bear physical meanings and can provide useful insights about the process. Moreover, comprehensive mechanistic models offer more predictive power given their deep connection to underlying molecular-scale transformations. In

chemical processing, the development of an accurate kinetic model is essential for optimal process design, plant safety, quality assurance, further analysis in the downstream, minimization of waste, and better control over troubleshooting. Therefore, complex kinetic models are an integral part of modern biological/chemical processes such as combustion,^{4–6} pyrolysis of biomass for alternative fuel production,^{7–9} catalytic conversions,^{10,11} cell metabolic processes,¹² and enzymatic reactions.¹³

Though a kinetic mechanism is essential to describe a complex chemical process, inaccurate descriptions of chemistry and poorly optimized kinetic parameters can cause a mechanism to fail or inaccurately describe the process. Typically, optimizing the kinetic parameters is the most challenging part as the number of parameters is usually high

Received: August 14, 2020

Revised: October 6, 2020

Accepted: October 8, 2020

Published: October 20, 2020



and they are extremely correlated within some uncertainty limits. At the same time, optimizing the model is not sensitive to all of the parameters; therefore, performing sensitivity analysis is required to list the most important parameter for both product and process optimization.¹⁴ In routine circumstances, when the experimental data is collected from a reactor and the reaction mechanism is connected with rate constants, thermodynamics, and the physical description of the reactor, usually a system of ordinary differential equations (ODEs) is generated. To perform parameter optimization and, often, any types of sensitivity analyses, it is required to solve the system of ODEs thousands of times. This imposes additional challenges in model optimization as these systems of ODEs are very stiff, which usually comes from disparate reaction rates between elementary steps, e.g., radical intermediate formation rates or termination steps. Therefore, an efficient stiff ODE solver is required to accelerate the model development process. Thus, developing a mechanistic kinetic model often consists of the following few basic steps: (i) formulation of the reaction mechanism to describe the chemistry, (ii) identification of the rate parameters and thermodynamic quantities from the literature or through quantum chemistry calculation, (iii) finding an efficient stiff ODE solver to accelerate model solution, (iv) performing sensitivity analysis to determine important model parameters, (v) optimizing model parameters to better describe experimental observations, and, finally, (vi) revising the initial model to better describe the chemistry. Furthermore, the mechanism also needs to be completed, which is often difficult to achieve because of multiplicity of mechanisms.¹⁵

Over the years, several databases of chemical reactions^{16,17} with kinetic parameters have been developed. In addition, with the help of efficient quantum chemistry software, it is possible to reasonably estimate the rate of elementary reactions. These initially estimated or obtained rate parameters can be a reasonable starting point to begin developing the kinetic model of interest; however, these might change based on the operating conditions and/or reactor types; therefore, parameter fitting or optimization is required. As mentioned previously, this is the most complex step of mechanistic modeling because the optimization depends on the initially estimated parameter values as they are highly correlated. Additionally, for gradient-based optimizers such as steepest gradient descent or conjugate gradient methods, the initial parameter choice(s) might lead the model to get trapped in a local minimum. Therefore, stochastic optimization methods have recently garnered more interest in use in optimizing chemical kinetic models for their ability to escape the local minima and converge to the global minima.

Stochastic optimization methods are general purpose optimization protocols, some of which are very flexible and can be applied to varieties of objective functions. The simplest one is the Monte Carlo (MC) method,¹⁸ where the objective function is evaluated over the entire parameter space. However, the number of required iterations increases exponentially with increasing numbers of parameters to optimize. Over the years, a good number of variants of stochastic or Monte Carlo algorithms have emerged, some of which even closely mimic efficient optimization approaches seen in nature. These various stochastic optimization algorithms have been used frequently in optimizing chemical kinetic models. For example, Vahteristo et al.¹⁹ used the Markov chain Monte Carlo (MCMC) method²⁰ to estimate

parameters for Diels–Alder reaction kinetics. Katare et al.²¹ used genetic algorithms (GAs),²² which is a method inspired by the natural selection process, to optimize large kinetic models. Several other works^{23–32} have reported the use of other stochastic methods like simulated annealing (SA),³³ particle swarm optimization (PSO),³⁴ differential evolution (DE),³⁵ shuffled complex evolution (SCE),³⁶ Latin hypercube sampling (LHS),³⁷ and robust parameter estimation (ROPE)³⁸ method. In addition, Da Ros et al.³⁹ compared various algorithms such as SA, PSO, DE, and artificial bee colony (ABC)⁴⁰ to estimate parameters for biochemical kinetic models and concluded that DE performs better compared to others. In contrast, Najari et al.⁴¹ compared DA and ABC and demonstrated that ABC is a more reliable algorithm to use. Since some algorithms contain tunable parameters to work efficiently, it is understandable that one algorithm might work better under certain parameter settings. This poses further challenges to the researchers as they need to tune multiple adjustable parameters before discovering which setting works better for their system. Although the above list is not exhaustive, we chose it for this study because it represents the set available methods with open source implementations available off-the-shelf in Python.

In spite of the development of a number of global stochastic optimizers, their relative performance in optimizing large chemical kinetic models is still debatable. The advent of free open source tools like Python makes many of these algorithms available nearly instantaneously, but comparative studies about their effectiveness for chemical kinetics problems are still needed. Since global optimization is itself a very difficult problem, kinetic model optimization usually starts with an already existing model, and new reactions are added to revise the model where rate parameters are estimated through quantum chemical calculations. Then, optimization is performed only for the parameters that are important or sensitive toward the experimental observations, which is used to fit the model. Therefore, though it is called global optimization, the number of parameters optimized is intentionally kept limited to maximize accuracy. For example, Siouris et al.⁴² optimized the Arrhenius rate parameters for a 21-step fuel degradation mechanism, while Ding et al.²⁴ optimized only 15 parameters to fully describe their lignocellulosic biomass pyrolysis process. However, a systematic study of how many parameters can be estimated simultaneously while optimizing the kinetic model and how the initial estimation of the model parameters affects the optimization quality is still missing, which will be of great importance for both process and product optimization. Thus, the focus of this study is to address both of these questions while comparing the performance of various stochastic optimizers and devise a methodology to successfully optimize a relatively large chemical kinetic model.

In this study, we systematically studied the performance of 10 different stochastic optimization algorithms, namely, Monte Carlo (MC), Latin hypercube sampling (LHS), maximum likelihood estimation (MLE),⁴³ Markov chain Monte Carlo (MCMC), shuffled complex evolution algorithm (SCE-UA), simulated annealing (SA), robust parameter estimation (ROPE), artificial bee colony (ABC), fitness scaled chaotic artificial bee colony (FSCABC),⁴⁴ and dynamically dimensioned search algorithm (DDS).⁴⁵ The system we consider is the San Diego mechanism⁴⁶ of propane oxidation, which has 46 species and 244 reactions; this mechanism is already

optimized and well known. In this study, we treated the distribution of various species concentrations generated from an already optimized San Diego mechanism as our training data, and then we randomized the rate parameters/constants to try to reproduce the observables using the stochastic algorithms mentioned above. It is important to mention here that some of the algorithms such as SA, ROPE, ABC, and FSCABC include adjustable parameters; however, we only consider a standard setting with default parameters or those suggested in the main documentation. Our main goal is to compare different algorithms, quantify their relative errors, and identify those algorithms that can optimize a significant number of parameters of a kinetic model to a reasonable accuracy but with minimum required adjustment of algorithmic parameters. After identifying a few candidate algorithms from initial screening, we present a detailed investigation to determine how well they perform.

Another obstacle in the field is the inability to reproduce the results of any of the many published kinetic models before one invests significant time and effort to rewrite the codes and troubleshooting them. Therefore, in the process of investigating the performance of various stochastic optimization algorithms, we have developed an open source Python package called kinexns,⁴⁷ which can handle the major steps of developing chemical kinetic models, namely, building and solving the system of ODEs for a simple closed system, performing sensitivity analysis, and optimizing kinetic parameters using various optimizers. Thus, an additional goal of this work was to publish the complete code and supporting files, as a well-documented, open source Python module that users with basic knowledge of Python can begin using immediately and start developing their own kinetic models.

2. CHEMICAL KINETIC MODEL AND COMPUTATIONAL DETAILS

2.1. Kinetic Model and ODE Solver. As mentioned previously, details of the reactions and kinetic parameters are taken from the San Diego mechanism⁴⁶ for propane oxidation. The input files were in Chemkin file format;⁴⁸ kinexns can parse mechanisms prepared in that format, converting them for further processing. For simplicity, we considered a perfectly mixed isothermal batch reactor; therefore, the model equations are straightforward ODEs for the concentrations of each species

$$\frac{dC_k}{dt} = R_k \quad (1)$$

where C_k is the concentration and R_k is the net rate of production of the k th species.

The rate of production can further be written as

$$R_k = \sum_{i=1}^I \nu_{ki} q_i \quad (2)$$

where I is the number of reactions in the mechanism and ν_{ki} is the stoichiometric coefficient of k th species in the i th reaction, which is further defined as

$$\nu_{ki} = \nu_{ki}' - \nu_{ki}'' \quad (3)$$

where the superscript ' indicates forward stoichiometric coefficients, while '' indicates reverse stoichiometric coefficients. The rate of progress variable q_i for the i th in eq 2 can further be defined as

$$q_i = k_{fi} \prod_{k=1}^K C_k^{\nu_{ki}'} - k_{ri} \prod_{k=1}^K C_k^{\nu_{ki}''} \quad (4)$$

where C_k is the molar concentration of the k th species and k_{fi} and k_{ri} are the forward and reverse rate constants of the i th reaction, respectively. These rate constants can be further calculated using the modified Arrhenius equation

$$k = AT^n \exp\left(-\frac{E_a}{RT}\right) \quad (5)$$

where A is the pre-exponential factor, T is the temperature, n is a constant, E_a is the activation energy, and R is the universal gas constant.

The mechanism we considered contains 46 individual species and 244 elementary reactions. Our python package automates the model building process and generates the system of ODEs ready to be solved once it reads all of the reactions and associated rate parameters. Due to the order of magnitude differences in both species formation rates and reaction rates, the system of ODEs is usually very stiff and requires an efficient ODE solver as the system needs to be solved thousands of times for both sensitivity analysis and optimization. At the same time, the solver has to be available open source so that we can use that in our python module. Therefore, we used the CVode ODE solver,⁴⁹ a solver for stiff and nonstiff ODE systems (initial value problem), developed at Lawrence Livermore National Laboratory as a part of SUNDIALS library.⁵⁰ CVode is also used in other multiscale software like OpenMKM⁵¹ for chemical kinetic modeling. In python, CVode solver is available through assimulo⁵² package, which has been integrated with kinexns. As suggested in the SUNDIALS website for stiff systems, the solution method was set to use backward differentiation formulas (BDFs) for the linear multistep method and Newton iteration for the nonlinear solver iteration. The user has control to set both the absolute and relative tolerance values; however, in case the solver fails to converge to a solution as the minimum allowable step size is reached, kinexns reduces the absolute tolerance by two orders of magnitude on the fly and tries to solve again until a solution is obtained.

2.2. Sensitivity Analysis. The first step in any kinetic model optimization is to identify the set of active parameters (i.e., the rate parameters that have the strongest impact on model outputs). For this reason, a global sensitivity analysis of the model parameters is required. Therefore, kinexns performs a variance-based global sensitivity analysis (Sobol sensitivity analysis)^{53–55} with the help of the SALib⁵⁶ python package. Sensitivity analysis can be performed on different model outputs (in our case, mostly the species concentrations) to various model inputs. Since we are interested in finding the reactions and their rate parameters to which the model outputs are highly sensitive to, in our case, the rate parameters were allowed to vary one order of magnitude above and below their specified values in the model at a specific temperature. Next, SALib, integrated with kinexns, can be used to generate a large set of parameter combinations that uniformly covers the entire parameter space. Once the simulations are performed over all of the parameter combinations, first-order, second-order, and total order sensitivity indices can be calculated through SALib. This procedure has also been followed in a previous publication from our lab.⁵⁷ Additionally, kinexns is equipped with generating bar plots for top n -sensitive reactions (can be

specified by the user) with sensitivity indices and drawing those reactions in a png file with the help of the rdkit⁵⁸ package to quickly visualize and identify the reactions. As mentioned previously, we used the Sobol sensitivity analysis, which is a global and model-independent sensitivity analysis method based on variance decomposition. Here, the total sensitivity indices (S_{Ti}) of a model parameter X_i are calculated as $S_{Ti} = S_i + \sum_{j \neq i} S_{ij}$, where S_i and S_{ij} are the first- and second-order Sobol sensitivity indices, respectively. For mathematical details, the users are referred to the original publications.^{55,59}

2.3. Optimization. Our main focus of this study is to test the performance of various stochastic algorithms to optimize kinetic models. Therefore, kinexns is integrated with an open source python package, spotpy,⁶⁰ which is a general purpose stochastic optimization module equipped with various stochastic algorithms. Spotpy was initially developed for optimizing environmental models, but it has the ability to be used for any optimization problem. The user has the control to choose which algorithm to use for optimization as well as which parameters to optimize. In kinexns, the user can decide either the reaction rate constants (at a specific temperature) or Arrhenius rate parameters (within a temperature range) to optimize. Also, the user can specify the reaction numbers whose rate constants/parameters are to be optimized.

Every optimization problem requires an objective/cost function to optimize; here, our cost function is to minimize the error between the model results and the actual values (generated by solving the model with the parameters from the San Diego mechanism) by using a weighted root-mean-square error (RMSE)⁶¹

$$\text{RMSE}_{\text{weighted}} = \sqrt{\sum_i^n \frac{1}{n} \left(\frac{y_{\text{actual}}[i] - y_{\text{model}}[i]}{0.1 y_{\text{actual}}[i]} \right)^2} \quad (6)$$

where RMSE is the root-mean-square error, y 's are the species concentrations in mol L⁻¹, and the index i refers to the set of all species, temperatures, and times for which fitting data is available. This formula assumes a relative error in each observation of 10% and calculates the residual accordingly to consider the fact that the experimentally obtained data contain some degree of error.

3. RESULTS AND DISCUSSION

Kinetic models are usually fit to some experimentally measurable data such as species concentrations. However, in this study, since we already have an optimized mechanism and our focus is to determine how accurately we can reproduce the results from that mechanism, we solved the model using the rate parameters listed in the San Diego mechanism and used the resulting data as our training data. Next, we used the concentration of a particular subset of species as our "training data" and used eq 6 as a cost function (training RMSE) to optimize the model. At the same time, we used the concentration data of all of the remaining species as our "test data" and calculated the weighted RMSE, which is referred to as the "test RMSE". We note for clarity that our definition of the train and test data for this study are somewhat different from conventional machine learning models where the entire data set is split into train and test data and both the train and test data have the same set of dependent variables/features. However, in our case, where we can loosely compare the rate constants with features, it is possible that the features

for one output in the test set (concentration of a particular species) are completely absent in the feature list of the training set, especially later in this study when we only optimized the sensitive reactions. Still, in our study, the test RMSE might give us an insight into how our fitted model works for the species that have not been used while optimizing the model.

3.1. Fivefold Cross-Validation. It is well known that our choice of training data can significantly affect both model and algorithm performance;^{62,63} therefore, it is necessary to find out how sensitive the algorithms are if we randomly choose our training data. To quantify this, we performed fivefold cross-validation with our training data to estimate how the results of relevant statistical analysis can be generalized given that the model is trained with different independent data sets. Here, we randomized our species (46 in total) into five equal but independent subsamples (nine species in each sample) and used the concentration of those species as our training data set and performed five different optimization tasks. Concentrations of all of the remaining species that were not included in training are used to calculate test RMSE. In this step, we only optimized the rate constants of all of the reactions in the mechanism (244 in total) at an arbitrarily chosen temperature of 950 K.

The training data was generated by solving the kinetic model with Arrhenius parameters listed in the San Diego mechanism at 950 K. Since we are studying propane combustion, we initialized our system with concentrations of 1 and 5 mol L⁻¹ for propane and oxygen, respectively, to create a stoichiometric mixture. Next, we randomized all of the rate constants calculated at 950 K within one order of magnitude above or below their prescribed value. Optimization of all 244 rate constants was performed using 10 different stochastic optimization algorithms in kinexns. We also note that there could be internal dependencies between these rate parameters. Although such a determination is out of the scope of this study, we note that a second-order sensitivity analysis would be helpful in extracting such relationships. Next, to perform fivefold cross-validation, five different optimization exercises are carried out for each algorithm with 10 000 optimization steps for each. The number of optimization steps is an adjustable parameter of the model, and later, we showed how our choice of 10 000 steps was enough to reasonably fit the model of interest.

Figure 1 shows a bar plot of the average RMSE values with standard deviations for fivefold cross-validation of all 10

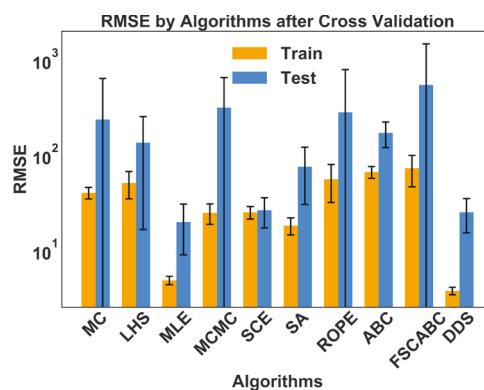


Figure 1. Average RMSE (log scale) with standard deviations for different algorithms after fivefold cross-validation.

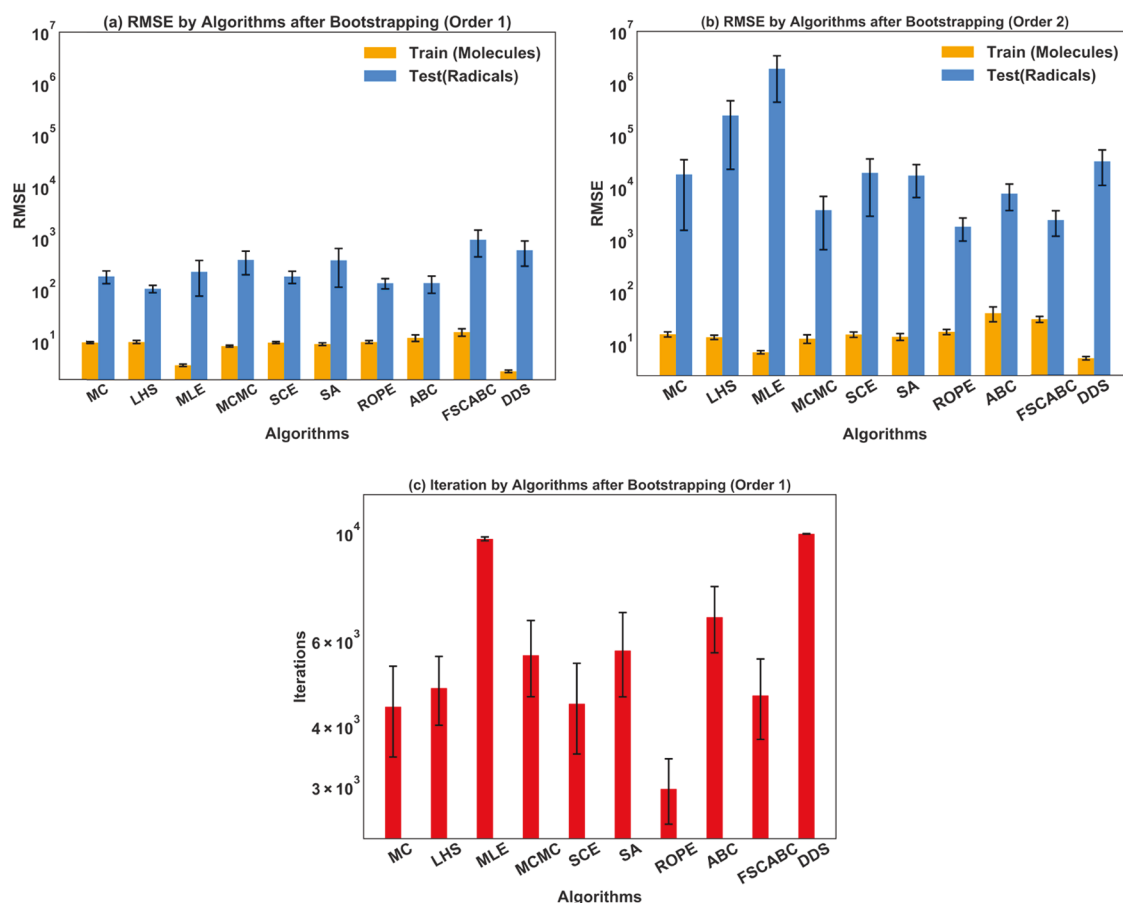


Figure 2. Average RMSE with standard deviations of train (molecules) and test (radicals) sets for different algorithms for (a) order 1 and (b) order 2 cases, and (c) average number of iterations required to optimize the objective function for different algorithms (order 1).

algorithms. It is clear from Figure 1 that DDS gives the lowest average RMSE with a very small standard deviation for training data, which is followed by MLE. The trend is similar for test RMSE as well, which is usually higher than training RMSE as our fitted models have never seen the test data. All other models except DDS and MLE perform more or less similar in terms of training RMSE, followed by SCE, SA, and MCMC. Also, in general, the standard deviation for training data is usually lower than that of test data, while for some cases (MC, MCMC, ROPE, and FSCABC), they are quite high for test data indicating that some of the fitted models might have converged to local minima; therefore, their test accuracy is very low. On the other hand, the low training RMSE values DDS and MLE could be a sign of strong overfitting; still, these two are performing reasonably well for test RMSE compared to other models. Therefore, we can conclude that both DDS and MLE are less sensitive to the choice of fitting data than the other algorithms in the set, and either algorithm could be a good choice for our application. Additionally, we observe that SCE has the best train to test the RMSE ratio, so it could also be a reasonable choice for our application.

3.2. Optimization of Rate Constants. The fivefold cross-validation results shown in the previous section demonstrate that algorithms like MLE and DDS do not depend on our choice of training data (i.e., the uncertainty on the mean training errors is very low). However, in the kinetic model optimization of complex systems, the radical concentrations are generally impossible to measure, and researchers usually use the concentrations of experimentally measured stable

molecules to fit their model. Therefore, as the next step, we decided to take the concentrations of all 19 stable molecules in the mechanism as our training data and concentration of all remaining 27 radical species as our test data (n.b., we assume to take the predicted radical species concentrations from the San Diego propane mechanism as the “ground truth”, recognizing that those values have not been fully validated with experimental observations). Additionally, in an effort to estimate the allowable variation in initial rate parameters for each algorithm, reaction rate constants were varied 1 and 2 orders of magnitude above and below of their prescribed value calculated at 950 K. These two cases are referred to as orders 1 and 2 below. To get a statistical estimate on the RMSE values, 10 different sets of initial guesses for all 244 rate constants are generated for each of the cases at 950 K, which is then optimized (also for 10 000 optimization steps for each) using the 10 different algorithms under consideration.

The average RMSE of both the cases (orders 1 and 2) for train and test data and number of iterations required to get the optimum values for order 1 case are shown in Figure 2. Here, we bootstrapped our RMSE data to better estimate the confidence interval. Bootstrapping is a powerful analysis tool that quantifies the uncertainties associated with any measured data set by increasing the number of data points through sampling with replacement. Through bootstrapping, we generated 500 estimates of data sets of a sample size of 10 for each case reported in Figure 2. All of the means and standard deviations reported in Figure 2 are calculated from the bootstrapped samples. The trends observed in Figure 2a,b

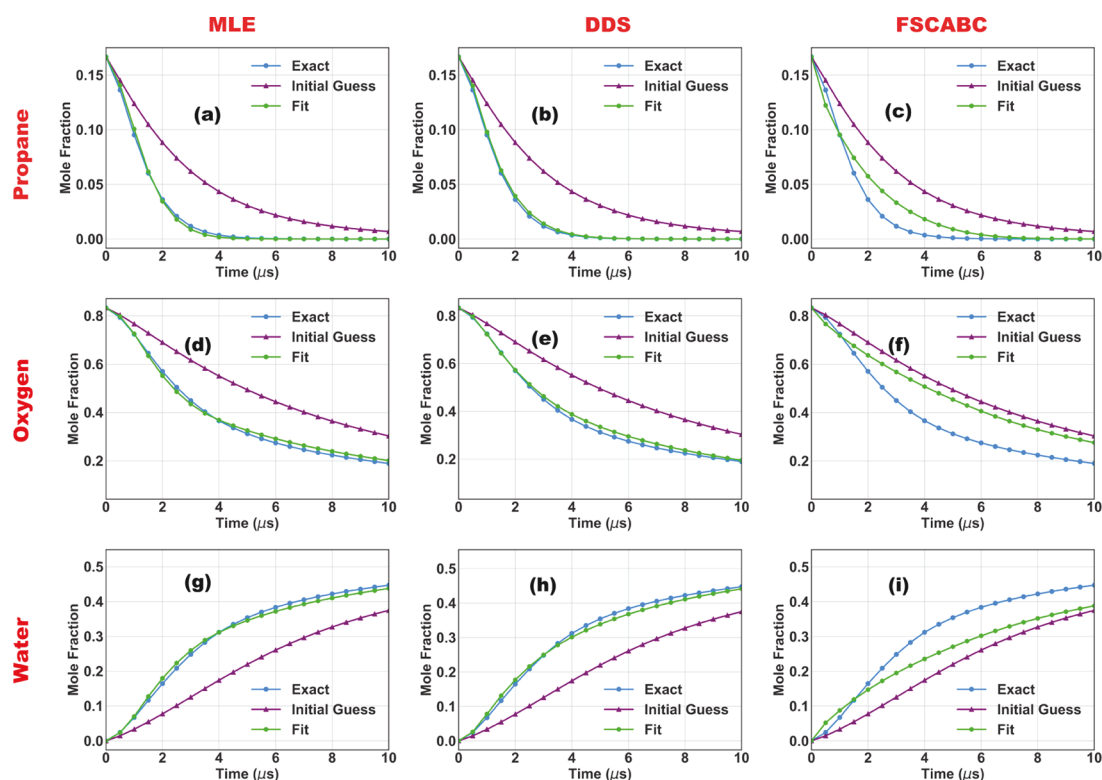


Figure 3. Distribution of species mole fraction (propane, oxygen, and water) before and after optimization using MLE, DDS, and FSCABC algorithms and comparison with the exact distribution for order 1 case. Each column represents an algorithm, while each row represents a species. (a) Algorithm: MLE, species: propane; (b) algorithm: DDS, species: propane; (c) algorithm: FSCABC, species: propane; (d) algorithm: MLE, species: oxygen; (e) algorithm: DDS, species: oxygen; (f) algorithm: FSCABC, species: oxygen; (g) algorithm: MLE, species: water; (h) algorithm: DDS, species: water; and (i) algorithm: FSCABC, species: water. Note that in this example a total of 244 parameters were optimized while using a total of 100 “experimental” data points for optimization (representing 19 species over the time series shown in the figure at a temperature of 950 K).

are very similar to those in Figure 1 as DDS and MLE give the lowest RMSE for training data, while ABC and FSCABC are being the highest. However, no particular trend is observed for test RMSE values in terms of algorithms. In fact, test RMSEs for MLE and DDS are very high for the order 2 case compared to the other algorithms. In general, test RMSE values for order 2 cases are a couple of magnitudes higher than the order 1 values, while no significant change in training RMSE is observed when an algorithm-wise comparison is made. This indicates that the algorithms (particularly DDS and MLE) still work well when the initial guesses are randomized over a broader range but only for training data sets. Since the test data set only contains the concentration of radicals that are impossible to measure experimentally, having a higher test RMSE does not alter our conclusion on the performance of the algorithms.

Figure 2c shows the average number of iterations required by each algorithm to optimize the cost function for order 1 case (the graph for order 2 case has not been shown as it is essentially similar). These numbers are obtained by getting the iteration number that reports the lowest RMSE value for the training set for each algorithm of 10 different simulations and then averaging them; here, no stopping criteria were used since some of the algorithms like DDS do not have the concept of stopping criteria. Iterations required by algorithms like MC, LHS, SCE, SA, ROPE, and FSCABC to report the lowest RMSEs are much less than the maximum iteration number (10 000) and vary within a broad range. This means these

algorithms have either become trapped in a local minimum after some while, which they fail to escape or already have converged their errors. However, DDS and MLE require almost always finding a better minimum but at a cost of longer training. Since there are no prescribed stopping criteria, these two algorithms will keep finding better minima in search of global minima as the simulation progresses. The nature of finding the minima by various algorithms can be explained more by how these algorithms work. Figure S1 in the Supporting Information shows the RMSE vs iteration for one sample run. Algorithms like MC, LHS, SCE, SA, ROPE, and FSCABC randomly evaluate the cost function using various parameter combinations within the parameter space and do not keep a memory of the previous best parameter combination, so it is easier for them to drift away from the global minima. However, DDS, MLE, and MCMC have an acceptance/rejection criterion and parameter combination only gets updated when the criterion is met—most of the time, this provides a better solution. Therefore, these algorithms are more likely to converge toward global minima. Also, this confirms that our choice of iteration number (10 000) was enough for all of the algorithms to converge to their respective minima.

As an additional performance metric, it is also important to visualize how closely the model solutions reproduce the ground truth data over the time series of the combustion model. Therefore, in Figure 3, we have plotted the optimized distribution profile of three major species (propane, oxygen,

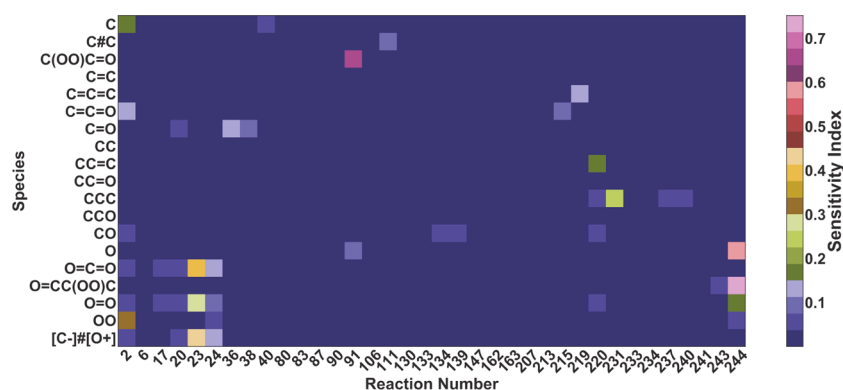


Figure 4. Sensitivity indices of the most sensitive reactions to model outputs. Here, the model outputs are the concentration of 19 different stable species. Top five sensitive reactions with the highest global sensitivity indices are listed for each model output; the model reduces to a set of 36 reactions only. The species are listed in the figure as smile strings.

and water) for order 1 case obtained using MLE, DDS, and FSCABC algorithms. As expected, the initial distributions shown in Figure 3 are within 1 order of magnitude of the exact solutions; this behavior is expected as the rate parameters are allowed to vary within 1 order of magnitude of the known values from the San Diego mechanism. A similar figure for order 2 case can be found in the Supporting Information (Figure S2). Our choice of algorithms in these figures is based on their performances as MLE and DDS are the best-performing ones and FSCABC is the worst.

Figure 3 shows that both MLE and DDS can almost identically reproduce the fitting (exact) distributions, which is expected from the previous discussion as they both report very low RMSE values. However, FSCABC fails to reproduce the exact distributions though the same initial distribution data was used. This is very much expected from our discussions related to Figure 2 as the average train RMSE resulted from fitting through FSCABC is very high. Similar trends can also be observed for order 2 case (Figure S2); however, the fitted distributions obtained from MLE and DDS are not as accurate as the order 1 case as the rate constants were allowed to vary more, which made the optimization task more complicated.

3.3. Sensitivity Analysis. The results presented in the above sections give us some initial idea about which stochastic algorithms might perform better in the kinetic model development; however, the method we presented is problematic in the sense that we only optimized the rate constants at a specific temperature while the kinetic models are usually developed over a temperature range. Thus, in spite of optimizing the rate constants, we need to optimize the Arrhenius rate parameters shown in eq 5.

Therefore, for each reaction, we need to optimize three parameters (A , n , and E_a) instead of one, which increases the numbers of parameters to optimize by threefold. As a result, performing sensitivity analysis on the model parameters is necessary to figure out the parameters sensitive from model output, consequently reducing the number of model parameters to optimize, thereby reducing the model complexity.

To determine the sensitive rate constants, which in turn will give us the sensitive reactions, we allowed the rate constants to vary 1 order of magnitude above and below their prescribed value. Our python package, kinexns, integrated with SALib⁵⁶ was used for the sensitivity analysis. At first, we generated 490 000 parameter combinations based on the quasi-random

Sobol sequence⁵⁵ that uniformly cover the entire possible input parameter space. Model outputs are simulated for each parameter combination. Therefore, we examined the sensitivity of all of the species concentration outputs (46 species in total) to all 244 rate constant values by calculating the variance-based total sensitivity indices. Figure S3 in the Supporting Information shows a heat map of sensitivity indices for the rate constants, which are evaluated to examine the sensitivity of the model outputs—the concentration of all of the species presented in the mechanism. The reaction numbers in Figure S3 and hereafter correspond to the order of appearance of the reactions in the input file; other than that, they bear no particular significance. Most of the sensitivity indices in Figure S3 are either zero or close to zero indicating that the specific model output is not sensitive to those reactions. Since we have already stated that the optimization of kinetic model is almost always performed on the experimentally measurable values such as species concentration, next, we limited our search for sensitive reactions to the concentration of stable molecules only, which are experimentally measurable. Therefore, we made a list of rate constants with the top five highest total sensitivity indices for the model outputs for each of the 19 stable species. This gave us a unique and reduced set of 36 reactions that the model output is sensitive to. The heat map of the sensitivity indices of the highly sensitive reactions for all 19 species is shown in Figure 4. Here, we would like to mention that no cutoff value was used to choose the sensitive reactions; we just listed the top five reactions sensitive to each stable species concentration. Therefore, we have ended up listing a few reactions that are not numerically highly sensitive to any of the species (i.e., reaction numbers 162 and 163 in Figure 4). If we had chosen a cutoff value, we could have further reduced the number of reactions.

3.4. Optimization of Arrhenius Rate Parameters for the Reduced Model. Now we have established a list of sensitive reactions, which reduced the model complexity to a great detail, the next step is to optimize the Arrhenius rate parameters instead of rate constants over a temperature range. For that purpose, we decided to randomize the Arrhenius rate parameters (A , n , and E_a) and fit the model data generated over a temperature range between 850 and 1050 K. Again, the choice of the temperature range is random as our goal is to determine how accurately we can fit the model data. Here, we calculated model solutions using the rate parameters prescribed in the San Diego mechanism for all of the reactions

from 850 to 1050 K with a temperature interval of 50 K and treated the results as our fitting data. Next, we varied the Arrhenius rate parameters ($\ln A$, n , and E_a) for the 36 sensitive reactions by $\pm 20\%$ from their prescribed value and calculated the rate constants at each temperature. The model was then solved five times (at five different temperatures) to calculate one RMSE. Similar to the previous section, we also calculated test RMSE considering the radical concentrations as well. Since the relationship between the Arrhenius rate parameters and the rate constants is not linear, we also looked at how much variation in rate constants we have introduced through our method, which is shown in Figure S4 in the Supporting Information section. As expected, the variation in rate constant values for the 36 sensitive reactions is not uniform, but most of them vary somewhere between ± 3 orders of magnitude from their base value; this variation in rate constants is even higher than that used for our order 2 case previously.

The quality of the fitting results for different algorithms after fitting with 10 different initial combinations of Arrhenius rate parameters is shown in Figure 5. Similar to that observed in

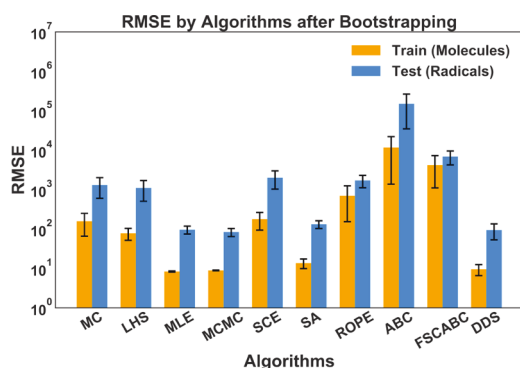


Figure 5. Average RMSE with standard deviations for different algorithms when Arrhenius rate parameters were optimized over a temperature range.

Figures 1 and 2, MLE and DDS work better here, while ABC and FSCABC perform the worst. However, some significant differences are observed when we closely compare Figure 5 with Figures 1 and 2. First, MCMC and SA start to perform much better than before when we optimized the rate parameters for sensitive reactions only; their train and test RMSEs are very similar to those achieved for MLE and DDS. This shows that both MCMC and SA can perform better when a more guided optimization protocol is followed. Second, there was no pattern between the train and test RMSEs previously, but in Figure 5, test RMSE follows the same pattern of train RMSE (i.e., the algorithms giving low train RMSE also give low test RMSE). Third, the overall test RMSE in Figure 5 for all of the algorithms is lower compared to that in Figure 2b, though the parameter variation is higher in the earlier. The reason behind the second and third differences lies in the optimization protocol we followed. Since we only optimized the rate parameters of the reactions that are sensitive to the species used to calculate train RMSE and kept all of the other parameters the same as the base model parameters, noise introduced initially in the test species is lower compared to that in our previous protocol. Here, we want to reiterate the fact that the test RMSE values bear much less significance as these species are not measurable experimentally; however, it

gives us confidence in our model solutions and helps us to evaluate the performance of the candidate algorithms.

In Figure 6, we compare the distribution of three major species when our model is fitted to optimize Arrhenius rate parameters for sensitive reactions and compared with the initial distribution and exact distribution for three different algorithms, MLE, DDS, and FSCABC, similar to Figure 3. The distribution showed here is calculated at 950 K. As we expected from the conclusion of Figure 5, MLE and DDS are still able to closely reproduce fitting data. An important feature to note about Figure 6 is that the initial distribution due to randomizing the rate parameters is very different from the exact distribution; still, both MLE and DDS are able to converge toward the optimum distribution. On the other hand, FSCABC is barely able to leave the initial distribution and poorly reproduces the exact distributions. Therefore, our observation that MLE and DDS are the best-performing algorithms still holds here, though the fitting quality is not as accurate as we observed in Figure 3.

The result presented so far shows that MLE and DDS consistently reproduce fitting data with greater accuracy, while FSCABC fails to estimate the parameters more and more as fitting complexity increases. Therefore, in Figure 7, we looked at the variation of the estimated parameters across these three algorithms. The result presented here is taken from 10 different optimization simulations started with 10 different initial sets of parameters. The rate parameters are varied $\pm 20\%$ from their prescribed value with a hope that they will converge to their original values upon optimization. Figure 7a shows the relative variation of the estimated $\ln A$, while Figure 7b shows the relative variation of the estimated E_a values after optimization. In an ideal case, all of the values shown in Figure 7 should converge to a 1.0 after optimization as they are the ratios of predicted to actual. However, in reality, the parameter values oscillate around 1.0. The range of the values estimated and their mean values can give us an idea of how each algorithm is doing. For all three algorithms, the mean values of the estimated parameters are more or less close to 1.0; however, in Figure 7a, the relative variation in estimating the values of $\ln A$ by FSCABC is almost always higher than that of the other two algorithms. It seems that FSCABC is estimating the $\ln A$ values within the entire range of parameter space, while both MLE and DDS are able to find their optimized value within a more confined space. This observation is not that pronounced in Figure 7b, where relative variation in estimating energy barriers is shown. Therefore, the failure of FSCABC to correctly estimate rate parameters to improve model accuracy mostly lies in its failure to predict the $\ln A$ values correctly—this makes complete sense as A is the most dominant parameter in calculating the rate constants and small variance in $\ln A$ can lead to drastically different rate constants as we have to take the exponential of $\ln A$ to calculate the rate constants.

Combining Figures 5 and 7, an interesting observation can be made in terms of global optimization. Namely, that several combinations of model parameters can result in similar RMSE values (i.e., there are many degenerate solutions). In some cases, the RMSE value is so high that a global optimization is not achieved. This clearly demonstrates the fact that the global optimization is not guaranteed with any algorithm and if there is a solution that might not be a unique one. It is also possible to slightly increase or decrease the already reduced parameter space and come up with a different global minimum. Therefore, it is still ambiguous that which model (i.e., set of

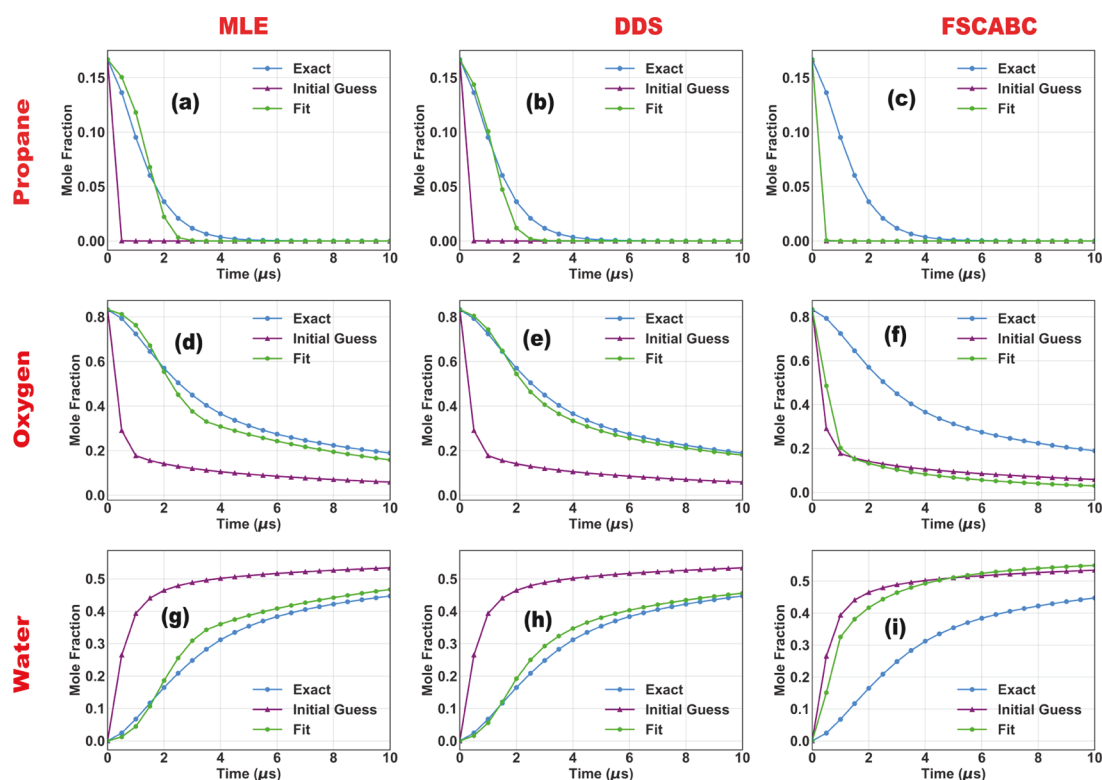


Figure 6. Distribution of species mole fraction (propane, oxygen, and water) before and after optimization using MLE, DDS, and FSCABC algorithms and comparison with the exact distribution when we optimized the Arrhenius rate parameters of the 36 sensitive reactions. Species distributions are plotted for 950 K. Each column represents an algorithm, while each row represents a species. (a) Algorithm: MLE, species: propane; (b) algorithm: DDS, species: propane; (c) algorithm: FSCABC, species: propane; (d) algorithm: MLE, species: oxygen; (e) algorithm: DDS, species: oxygen; (f) algorithm: FSCABC, species: oxygen; (g) algorithm: MLE, species: water; (h) algorithm: DDS, species: water; and (i) algorithm: FSCABC, species: water. In contrast to the fitting in Figure 3, this exercise optimized the Arrhenius rate parameters over a temperature range of 850–1050 K, not the Arrhenius rate constants at a specific temperature.

optimized parameters) is to be employed. To address these broader issues, it is important that the users carefully consider the application or domain of the kinetic modeling and hopefully identify multiple independent test/use cases for further testing the models.

The results presented so far consistently show that MLE and DDS are the best-performing algorithms for large-scale optimization of kinetic models; MCMC follows next. However, researchers have used other algorithms such as LHS, SCE, ROPE, and ABC successfully in kinetic model optimization.^{28,30,32,41,64} Therefore, it requires an explanation of why our model fails to perform with these algorithms. If we look at how the algorithms work, it reveals two important features of the best-performing algorithms that we think could be responsible for such behavior. First, algorithms like MLE, DDS, and MCMC are hyperparameter-free optimizers. It means that they can perform optimization under a universal parameter setting that does not depend on the model or the cost function. On the other hand, algorithms like SA require parameters such as annealing scheduling temperature and initial temperature to be specified before starting the optimization. ABC requires parameters like number of generations and maximum number of trials for abandoning food source to be prespecified. These parameters are called hyperparameters and may need different settings for different problems. Therefore, these parameters need to be optimized based on the system before an overall model optimization can be performed; this is also a routine work in developing

machine learning models where the hyperparameters are optimized via grid search. Here, we have only used the default hyperparameter settings for these algorithms made available in spotpy. Interested readers can go through the spotpy documentation (https://spotpy.readthedocs.io/en/latest/Advanced_hints/) to learn more about the settings and how to change them. We believe that, with proper hyperparameter settings, SCE, ROPE, ABC, and FSCABC might be able to estimate model parameters more accurately; however, this adds an extra layer of complexity to optimize hyperparameters in an already complicated problem (i.e., a problem in which there are substantial computational costs generating the model simulation results); therefore, hyperparameter-free optimizers are always preferred. Second, MLE, DDS, and MCMC are “greedy”-type algorithms; it means that they have an acceptance/rejection criterion. For these algorithms, the current solution, also the best solution identified so far, is only updated with a solution that has met the acceptance/rejected criteria. If the cost function RMSE found in the next iteration is lower than the current one, the solution is always accepted. On the other hand, if it is higher, the solution is only accepted if they are within some probability limit. This enables the ability to escape the local minima and at the same time keeps the track of previously found best solutions. Therefore, these algorithms have a better chance to escape local minima and converge to the global one; however, it will involve more optimization steps.

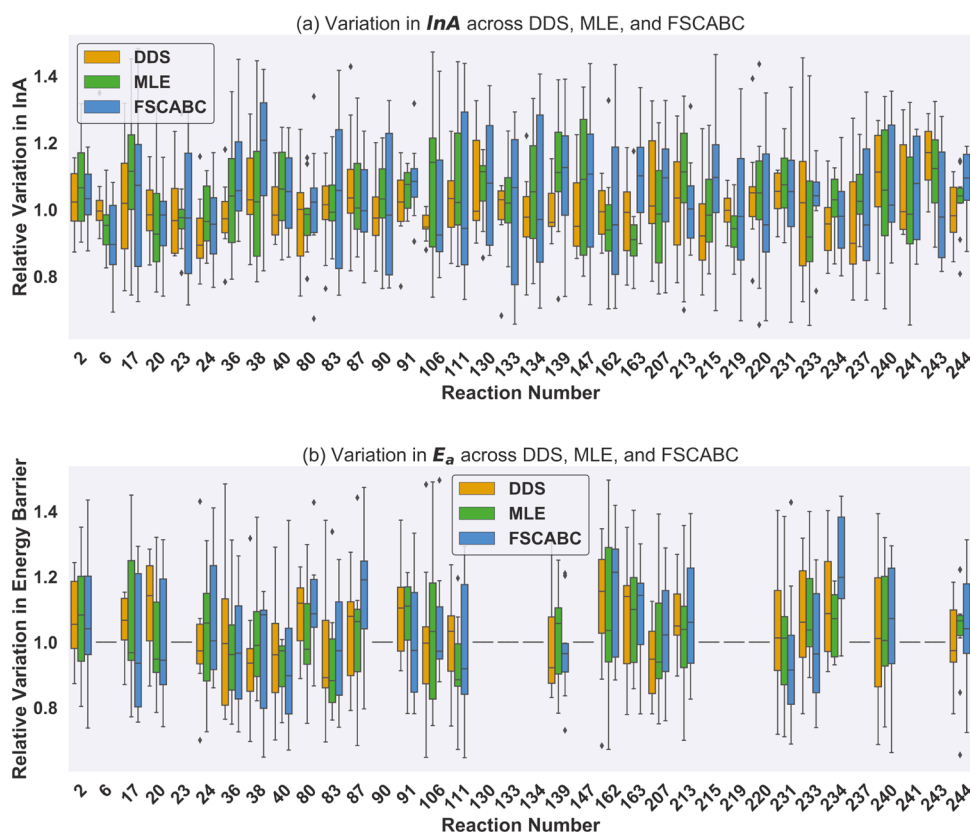


Figure 7. Variation in reaction rate parameters after optimization through MLE, DDS, and FSCABC. (a) Relative variation in estimating $\ln A$ and (b) relative variation in estimating E_a . Variation is reported from 10 different simulations with 10 different initial parameter sets to optimize.

As a final analysis, we explored how the training RMSE changes when we systematically increase the number of species used to train this model. Here, we have chosen only four different algorithms instead of all 10, namely, MLE, MCMC, SA and DDS, as we have already shown that these are the top-performing algorithm for our model. For this analysis, we choose different numbers of stable species (3, 6, 9, 12, 15, and 19) whose concentration will be used to train the model. The choice of species in each system is random. Next, we performed sensitivity analysis on the model outputs of the chosen species to create a list of reactions (top five reactions, which each species output is sensitive to were chosen); the rate parameters of these reactions are used only for the training. Table 1 shows the number of reactions included for each case;

Table 1. Number of Species Included in the Model vs Number of Reactions to Optimize

no. of species	3	6	9	12	15	19
no. of reactions	12	15	22	28	34	36

with increasing number of species to train the model, the number of reactions to optimize also increases but the relation is not exactly linear. Next, we optimized all of the cases for five different initial parameter sets; all other procedures of optimization remain the same.

Figure 8a shows that, for all four algorithms, training RMSE increases linearly with the number of species used to train the model. When the number of species is low, the training RMSE of all of the algorithms is very low; this is expected, as the model is optimized to reproduce only a small number of species concentration. At the same time, the number of

parameters to optimize is low too. As the number of species increases, though all of the training RMSE increases, a diverging tendency between the models is observed. The increase in RMSE values with increased number of species is due to two main reasons: first, the number of parameters to optimize increases, which makes the optimization process difficult, and second, the relationship between the number of species to optimize and the number of sensitive reactions is not linear (Table 1), which leads to fewer data points per species to optimize. RMSEs obtained from MCMC and SA optimization start to slightly diverge from RMSEs obtained from MLE and DDS. This indicates that, as the number of parameters to optimize starts to increase, the performance of MCMC and SA algorithms starts to deteriorate more quickly compared to that of MLE and DDS. Therefore, we can finally conclude that MLE and DDS are the best-performing stochastic algorithms that can be used to optimize complex chemical models with great complexity.

Finally, we wanted to understand how the training RMSE changes if we include all of the 19 stable species to train, but instead of optimizing only the parameters of the most sensitive reactions, we try to optimize all of the parameters of the model (244 reactions). Figure 8b compares the training RMSE of the reduced and complete mechanism. For all of the four algorithms considered, it is observed that training RMSE increases slightly when the complete mechanism is optimized instead of the reduced mechanism. This demonstrates the importance of sensitivity analysis. First, since the model is not equally sensitive to all of the parameters, it is not required to optimize all of the model parameters at the same time. This, in turn, reduces the model complexity. Second, though both the

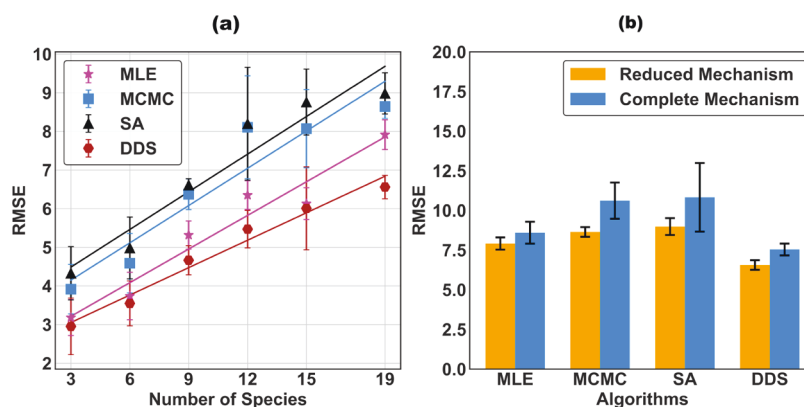


Figure 8. (a) Variation of RMSE with the number of species included in the model to train and (b) comparison of RMSE values between the reduced (36 reactions) and complete (244 reactions) mechanism. Results are only shown for MLE, MCMC, SA, and DDS algorithms.

optimization exercise (36 reactions vs 244 reactions) use the same fitting, the training RMSE for the 244-reaction case is always higher as it is trying to optimize a greater number of parameters, although the model is not sensitive to a majority of those. Therefore, by choosing the parameters that the model is sensitive to and optimizing only those parameters, one can achieve higher accuracy in the model rather than optimizing all of the parameters in a single go. This clearly demonstrates the requirement of sensitivity analysis before performing any optimization.

One of the major concerns of these stochastic methods over the gradient-based method is the increased computational cost compared to gradient-based ones that use simple numerical differentiation to optimize. However, there is a strong possibility that the gradient-based optimization will be trapped in a local minimum; therefore, it is used for these types of optimizations only when there is high confidence in the initial parameter guesses. To provide some context for readers, we make some general estimates of computational costs. The computational cost of this study can be divided into two parts: solving the ODEs and optimizing the loss functions. The former is the same for all of the algorithms and usually took ~ 0.6 – 0.7 s for each iteration (simulations were run on the UW Hyak supercomputer cluster (Intel(R) Xeon(R) Gold 6230 CPU@2.10 GHz machine)). Also, in Table S1 in the Supporting Information, we have tabulated the time required to complete 100 iterations for all of the algorithms, which includes both solving the ODEs and optimizing the loss functions. Since for all of the cases the system of ODEs is solved 100 times each, the discrepancy in time comes from the optimization part where SCE-UA and LHS are the fastest and SA is the slowest. However, multiprocessing is implemented into kinexns, which allows us to run all of those optimization algorithms simultaneously (multicore, single node), which reduces the computation time significantly.

Finally, though some of the stochastic methods discussed in this manuscript performed reasonably well for the tests we performed for propane oxidation mechanism, we have to keep in mind that this mechanism is on the smaller side with only 46 species and 244 reactions. Reaction mechanisms for larger hydrocarbons and fuel mixtures are more complicated with hundreds of species and thousands of reactions,⁶⁵ which presents additional challenges as more parameters need to be optimized. At this point, it is still to be determined how these stochastic optimization algorithms will handle this situation and with what level of robustness. This is especially true when

the number of parameters for global optimization exceeds the available training data. As this is often the case for global optimization of complex mechanisms, and researchers have been using a few alternatives to reduce the parameter space. One of those alternatives, sensitivity analysis to determine model-sensitive parameters, is also used in this study. Another frequently used method is combining reactions based on their classes known by reaction class methodology and proposed by Curran et al.^{66,67} and Sarathy et al.⁶⁸ As mentioned, both of these methods will help to develop a reduced mechanism with smaller parameter space to optimize.

4. CONCLUSIONS

Process optimization lies at the center of any industrial bio/chemical process, and accurate, efficient optimization methods are required to successfully estimate the model parameters. Here, we presented a comprehensive study on how different stochastic optimizers work when they are subjected to the same optimization task. At the same time, we suggested some best practices required for large-scale kinetic model optimization, hopefully beneficiary new entrants in chemical kinetic modeling. Among the 10 different algorithms we studied, two algorithms, namely, MLE and DDS, performed consistently better over all other algorithms; MCMC and SA followed them in terms of accuracy when a systematic optimization procedure was introduced through sensitivity analysis. On the other hand, ABC and FSCABC were always the worst ones, though previous studies have shown that upon setting correct hyperparameters, they are also able to estimate model parameters with reasonable accuracy.^{39,41}

One major outcome of this study is to demonstrate that the hyperparameter-free optimizers such as MLE and DDS are the best-performing algorithms for chemical kinetic modeling, which can globally optimize a high number of model parameters with reasonable accuracy even when the initial guess values are far from the optimized ones. This will reduce the load from the modelers' side significantly as they do not have to optimize the hyperparameters for the algorithms. Hyperparameter setting usually varies from model to model; therefore, hyperparameter-free algorithms would be easily transferable when a different system is to be optimized. Also, this is the first time that the use of DDS algorithm to optimize a chemical kinetic model is demonstrated. DDS was initially optimized for efficient watershed modeling,⁴⁵ and the authors demonstrated that DDS requires only 15–20% of the number of model evaluations compared to SCE to find equally good

value for the objective function. Here, we demonstrated that this method can be successfully used in optimizing chemical kinetic models as well. Similarly, there are limited examples of MLE optimizers for the application of chemical kinetic models in the literature. Additionally, with proper hyperparameter settings, algorithms like MCMC, SA, and ABC could potentially perform much better than reported in this study and we propose that future efforts should consider computationally efficient and systematic procedures for hyperparameter tuning.

A major challenge in this field is the difficulty in reproducing and utilizing published models. Over the past few decades, numerous kinetic models have been developed; however, either the documentations provided in the literature are not clear enough or the codes written to solve the proposed model are not made available to quickly and successfully reproduce the results. Reconstruction of the models can take weeks as a lot of coding and debugging is required. Also, researchers might have to change their platform to perform sensitivity analysis and model optimization, the two most important aspects of model generation. Thus, another major aspect of this study was to develop an open source python package, kinexns, that can solve the model efficiently, identify sensitive reactions/model parameters, and perform model optimization. We have published the entire code base on Github and have made it available to other researchers. The code is written in modules and can read Chemkin input files. All of the model building processes (building the system of ODEs) are automated so that the user can start using the package with relative ease. Also, the user can add his/her own features easily with the existing code base (e.g., the package can now only handle batch reactors; however, it can be easily expanded to solve semibatch reactor models if a new module is added). We hope the ease of using this package will facilitate the improvement and expansion of existing kinetic models while developing new ones. This package could also be a good resource to use in classroom teaching of building and solving chemical kinetic models.

■ ASSOCIATED CONTENT

SI Supporting Information

The Supporting Information is available free of charge at <https://pubs.acs.org/doi/10.1021/acs.iecr.0c04009>.

Propagation of RMSE for various algorithms with iterations (Figure S1), distribution of species mole fraction for order 2 case (Figure S2, constructed similarly as Figure 3), sensitivity indices of all reactions (Figure S3), maximum variation of reaction rate constants when the model is optimized over a temperature range (Figure S4); all of the codes developed can be found at <https://github.com/cmashraf/kinexns>; 47 example jupyter notebooks demonstrating the complete workflow with sample input files can be found at <https://github.com/cmashraf/kinexns/tree/master/Notebooks> (PDF)

■ AUTHOR INFORMATION

Corresponding Author

Jim Pfaendtner – Department of Chemical Engineering,
University of Washington, Seattle, Washington 98195, United States; orcid.org/0000-0001-6727-2957;
Email: jpfaendt@uw.edu

Author

Chowdhury Ashraf – Department of Chemical Engineering,
University of Washington, Seattle, Washington 98195, United States; orcid.org/0000-0002-7375-2540

Complete contact information is available at:
<https://pubs.acs.org/10.1021/acs.iecr.0c04009>

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This research was supported by NSF award #1703638. C.A. would also like to acknowledge the support of the University of Washington eScience Institute through an award from the Washington Research Foundation. All of the simulations were performed in the Hyak supercomputer system at the University of Washington. The authors would like to thank Luke Gibson for his careful reading of the manuscript and helpful discussions.

■ REFERENCES

- (1) Huang, C.; Li, S.; Wang, J.; Yang, B.; Zhang, F. Global Uncertainty Analysis for the RRKM/Master Equation Modeling of a Typical Multi-Well and Multi-Channel Reaction System. *Combust. Flame* **2020**, *216*, 62–71.
- (2) Wang, H.; Xu, R.; Bowman, C. T.; Hanson, R. K.; Davidson, D. F.; Brezinsky, K.; Egolfopoulos, F. N. A Physics-Based Approach to Modeling Real-Fuel Combustion Chemistry - I. Evidence from Experiments, and Thermodynamic, Chemical Kinetic and Statistical Considerations. *Combust. Flame* **2018**, *193*, S02–S19.
- (3) Wang, H.; Sheen, D. A. Combustion Kinetic Model Uncertainty Quantification, Propagation and Minimization. *Prog. Energy Combust. Sci.* **2015**, *47*, 1–31.
- (4) Frenklach, M. Systematic Optimization of a Detailed Kinetic Model Using a Methane Ignition Example. *Combust. Flame* **1984**, *58*, 69–72.
- (5) Davis, S. G.; Joshi, A. V.; Wang, H.; Egolfopoulos, F. An Optimized Kinetic Model of H₂/CO Combustion. *Proc. Combust. Inst.* **2005**, *30*, 1283–1292.
- (6) Simmie, J. M. Detailed Chemical Kinetic Models for the Combustion of Hydrocarbon Fuels. *Prog. Energy Combust. Sci.* **2003**, *29*, 599–634.
- (7) Wang, S.; Dai, G.; Yang, H.; Luo, Z. Lignocellulosic Biomass Pyrolysis Mechanism: A State-of-the-Art Review. *Prog. Energy Combust. Sci.* **2017**, *62*, 33–86.
- (8) Ranzi, E.; Debiagi, P. E. A.; Frassoldati, A. Mathematical Modeling of Fast Biomass Pyrolysis and Bio-Oil Formation. Note I: Kinetic Mechanism of Biomass Pyrolysis. *ACS Sustainable Chem. Eng.* **2017**, *5*, 2867–2881.
- (9) Wang, S.; Lin, H.; Ru, B.; Dai, G.; Wang, X.; Xiao, G.; Luo, Z. Kinetic Modeling of Biomass Components Pyrolysis Using a Sequential and Coupling Method. *Fuel* **2016**, *185*, 763–771.
- (10) Matera, S.; Schneider, W. F.; Heyden, A.; Savara, A. Progress in Accurate Chemical Kinetic Modeling, Simulations, and Parameter Estimation for Heterogeneous Catalysis. *ACS Catal.* **2019**, *9*, 6624–6647.
- (11) Stempien, J. P.; Ni, M.; Sun, Q.; Chan, S. H. Thermodynamic Analysis of Combined Solid Oxide Electrolyzer and Fischer–Tropsch Processes. *Energy* **2015**, *81*, 682–690.
- (12) Bordbar, A.; McCloskey, D.; Zielinski, D. C.; Sonnenschein, N.; Jamshidi, N.; Palsson, B. O. Personalized Whole-Cell Kinetic Models of Metabolism for Discovery in Genomics and Pharmacodynamics. *Cell Syst.* **2015**, *1*, 283–292.
- (13) Rios-Solis, L.; Morris, P.; Grant, C.; Odeleye, A. O. O.; Hailes, H. C.; Ward, J. M.; Dalby, P. A.; Bagan, F.; Lye, G. J. Modelling and Optimisation of the One-Pot, Multi-Enzymatic Synthesis of Chiral

Amino-Alcohols Based on Microscale Kinetic Parameter Determination. *Chem. Eng. Sci.* **2015**, *122*, 360–372.

(14) vom Lehn, F.; Cai, L.; Pitsch, H. Sensitivity Analysis, Uncertainty Quantification, and Optimization for Thermochemical Properties in Chemical Kinetic Combustion Models. *Proc. Combust. Inst.* **2019**, *37*, 771–779.

(15) Harold, M. P.; Luss, D. Use of Multiplicity Features for Kinetic Modeling: Carbon Monoxide Oxidation on Platinum/Alumina. *Ind. Eng. Chem. Res.* **1987**, *26*, 2099–2106.

(16) ChemWeb Databases. <https://www.chemweb.com/databases>.

(17) Reaction Databases/Organic Chemistry Resources Worldwide. <http://www.organicworldwide.net/content/reaction-databases>.

(18) Metropolis, N.; Ulam, S. The Monte Carlo Method. *J. Am. Stat. Assoc.* **1949**, *44*, 335–341.

(19) Vahteristo, K.; Laari, A.; Solonen, A. Diels–Alder Reaction Kinetics for Production of Norbornene Monomers: Evaluation of Parameters and Model Reliability Using Markov Chain Monte Carlo Methods. *Ind. Eng. Chem. Res.* **2013**, *52*, 6357–6365.

(20) Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H.; Teller, E. Equation of State Calculations by Fast Computing Machines. *J. Chem. Phys.* **1953**, *21*, 1087–1092.

(21) Katare, S.; Bhan, A.; Caruthers, J. M.; Delgass, W. N.; Venkatasubramanian, V. A Hybrid Genetic Algorithm for Efficient Parameter Estimation of Large Kinetic Models. *Comput. Chem. Eng.* **2004**, *28*, 2569–2581.

(22) Holland, J. H. *Adaptation in Natural and Artificial Systems*. University of Michigan Press: Ann Arbor, 1975; Vol. 1, p 975.

(23) dos Santos Rocha, M. S. R.; Pratto, B.; de Sousa, R.; Almeida, R. M. R. G.; da Cruz, A. J. G. A Kinetic Model for Hydrothermal Pretreatment of Sugarcane Straw. *Bioresour. Technol.* **2017**, *228*, 176–185.

(24) Ding, Y.; Ezekoye, O. A.; Zhang, J.; Wang, C.; Lu, S. The Effect of Chemical Reaction Kinetic Parameters on the Bench-Scale Pyrolysis of Lignocellulosic Biomass. *Fuel* **2018**, *232*, 147–153.

(25) Prata, D. M.; Schwaab, M.; Lima, E. L.; Pinto, J. C. Simultaneous Robust Data Reconciliation and Gross Error Detection through Particle Swarm Optimization for an Industrial Polypropylene Reactor. *Chem. Eng. Sci.* **2010**, *65*, 4943–4954.

(26) Parsopoulos, K. E.; Vrahatis, M. N. Recent Approaches to Global Optimization Problems through Particle Swarm Optimization. *Nat. Comput.* **2002**, *1*, 235–306.

(27) Bhat, T. R.; Venkataramani, D.; Ravi, V.; Murty, C. V. S. An Improved Differential Evolution Method for Efficient Parameter Estimation in Biofilter Modeling. *Biochem. Eng. J.* **2006**, *28*, 167–176.

(28) Ding, Y.; Wang, C.; Chaos, M.; Chen, R.; Lu, S. Estimation of Beech Pyrolysis Kinetic Parameters by Shuffled Complex Evolution. *Bioresour. Technol.* **2016**, *200*, 658–665.

(29) Liu, H.; Wang, C.; Chen, B.; Zhang, Z. A Further Study of Pyrolysis of Carbon Fibre-Epoxy Composite from Hydrogen Tank: Search Optimization for Kinetic Parameters via a Shuffled Complex Evolution. *J. Hazard. Mater.* **2019**, *374*, 20–25.

(30) Park, J.; Lee, K. S.; Kim, M. S.; Jung, D. Numerical Analysis of a Dual-Fueled CI (Compression Ignition) Engine Using Latin Hypercube Sampling and Multi-Objective Pareto Optimization. *Energy* **2014**, *70*, 278–287.

(31) Zhukov, V. V.; Laari, A.; Koironen, T. Kinetic Modeling of High-Pressure Pyrite Oxidation with Parameter Estimation and Reliability Analysis Using the Markov Chain Monte Carlo Method. *Ind. Eng. Chem. Res.* **2015**, *54*, 9920–9930.

(32) Krishnan, S.; Barton, G. W.; Perkins, J. D. Robust Parameter Estimation in On-Line Optimization—Part 2. Application to an Industrial Process. *Comput. Chem. Eng.* **1993**, *17*, 663–669.

(33) Szu, H.; Hartley, R. Fast Simulated Annealing. *Phys. Lett. A* **1987**, *122*, 157–162.

(34) Kennedy, J.; Eberhart, R. In *Particle Swarm Optimization (PSO)*, *Proceedings of the IEEE International Conference on Neural Networks*, Perth, Australia, 1995; pp 1942–1948.

(35) Storn, R. In *On the Usage of Differential Evolution for Function Optimization*, *Proceedings of North American Fuzzy Information Processing*, 1996; pp 519–523.

(36) Duan, Q. Y.; Gupta, V. K.; Sorooshian, S. Shuffled Complex Evolution Approach for Effective and Efficient Global Minimization. *J. Optim. Theory Appl.* **1993**, *76*, 501–521.

(37) Huntington, D. E.; Lyrantzis, C. S. Improvements to and Limitations of Latin Hypercube Sampling. *Probab. Eng. Mech.* **1998**, *13*, 245–253.

(38) Stewart, C. V. Robust Parameter Estimation in Computer Vision. *SIAM Rev.* **1999**, *41*, 513–537.

(39) Da Ros, S.; Colusso, G.; Weschenfelder, T. A.; de Marsillac Terra, L.; de Castilhos, F.; Corazza, M. L.; Schwaab, M. A Comparison among Stochastic Optimization Algorithms for Parameter Estimation of Biochemical Kinetic Models. *Appl. Soft Comput.* **2013**, *13*, 2205–2214.

(40) Karaboga, D.; Basturk, B. A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm. *J. Global Optim.* **2007**, *39*, 459–471.

(41) Najari, S.; Gróf, G.; Saeidi, S.; Gallucci, F. Modeling and Optimization of Hydrogenation of CO₂: Estimation of Kinetic Parameters via Artificial Bee Colony (ABC) and Differential Evolution (DE) Algorithms. *Int. J. Hydrogen Energy* **2019**, *44*, 4630–4649.

(42) Siouris, S.; Blakey, S. Fitness Functions for Evolutionary Optimization of Rate Parameters in Chemically Reacting Systems. *Chem. Eng. Sci.* **2019**, *196*, 354–365.

(43) Maximum Likelihood Estimation of the Parameters of a Normal Distribution which is Truncated at a Known Point: Scandinavian Actuarial Journal: Vol 1949, No 1. <https://www.tandfonline.com/doi/abs/10.1080/03461238.1949.10419767> (accessed Dec 1, 2019).

(44) Zhang, Y.; Wu, L.; Wang, S. UCAV Path Planning Based on FSCABC. *Inf.-Int. Interdiscip. J.* **2011**, *14*, 687–692.

(45) Tolson, B. A.; Shoemaker, C. A. Dynamically Dimensioned Search Algorithm for Computationally Efficient Watershed Model Calibration. *Water Resour. Res.* **2007**, *43*, No. 635.

(46) Chemical Mechanism: Combustion Research Group at UC San Diego. <https://web.eng.ucsd.edu/mae/groups/combustion/mechanism.html> (accessed Dec 2, 2019).

(47) Ashraf, C.; Pfaendtner, J. Kinexns (Kinetics of Reactions): An Open Source Python Package for Chemical Kinetic Modeling, Sensitivity Analysis and Optimization; Zenodo, 2020. <https://doi.org/10.5281/zenodo.3979937>.

(48) Kee, R. J.; Rupley, F. M.; Meeks, E.; Miller, J. A. *CHEMKIN-III: A FORTRAN Chemical Kinetics Package for the Analysis of Gas-Phase Chemical and Plasma Kinetics*; Sandia National Labs.: Livermore, CA, 1996.

(49) Cohen, S. D.; Hindmarsh, A. C.; Dubois, P. F. CVODE, a Stiff/Nonstiff ODE Solver in C. *Comput. Phys.* **1996**, *10*, 138–143.

(50) SUNDIALS: SUite of Nonlinear and Differential/ALgebraic Equation Solvers. <https://computing.llnl.gov/projects/sundials> (accessed Dec 2, 2019).

(51) Medasani, B.; Frey, J.; Lym, J. VlachosGroup/Openmkm: OpenMKM First Public Release; Zenodo, 2019. <https://doi.org/10.5281/zenodo.3532034>.

(52) Andersson, C.; Führer, C.; Åkesson, J. Assimulo: A Unified Framework for ODE Solvers. *Math. Comput. Simul.* **2015**, *116*, 26–43.

(53) Saltelli, A. Making Best Use of Model Evaluations to Compute Sensitivity Indices. *Comput. Phys. Commun.* **2002**, *145*, 280–297.

(54) Saltelli, A.; Annoni, P.; Azzini, I.; Campolongo, F.; Ratto, M.; Tarantola, S. Variance Based Sensitivity Analysis of Model Output. Design and Estimator for the Total Sensitivity Index. *Comput. Phys. Commun.* **2010**, *181*, 259–270.

(55) Sobol, I. M. Global Sensitivity Indices for Nonlinear Mathematical Models and Their Monte Carlo Estimates. *Math. Comput. Simul.* **2001**, *55*, 271–280.

- (56) Herman, J. D.; Usher, W. SALib: An Open-Source Python Library for Sensitivity Analysis. *J. Open Source Softw.* **2017**, *2*, No. 97.
- (57) Hough, B. R.; Schwartz, D. T.; Pfaendtner, J. Detailed Kinetic Modeling of Lignin Pyrolysis for Process Optimization. *Ind. Eng. Chem. Res.* **2016**, *55*, 9147–9153.
- (58) Landrum, G.; Tosco, P.; Kelley, B.; sriniker; gedec; Schneider, N.; Vianello, R.; Ric; Dalke, A.; Cole, B.; Alexander, S.; Turk, S.; Swain, M.; Dan, N.; Vaucher, A.; Wójcikowski, M.; Kawashima, E.; Pahl, A.; JP; Berenger, F.; strets123; Varjo, J. L.; O'Boyle, N.; Cosgrove, D.; Fuller, P.; Jensen, H.; Sforza, G.; Gavid, D.; Leswing, K.; van Santen, J. Rdkit/Rdkit:2019_09_1 (Q3 2019) Release; Zenodo, 2019. <https://doi.org/10.5281/zenodo.3518540>.
- (59) Nossent, J.; Elsen, P.; Bauwens, W. Sobol' Sensitivity Analysis of a Complex Environmental Model. *Environ. Model. Softw.* **2011**, *26*, 1515–1525.
- (60) Houska, T.; Kraft, P.; Chamorro-Chavez, A.; Breuer, L. SPOTting Model Parameters Using a Ready-Made Python Package. *PLoS One* **2015**, *10*, No. e0145180.
- (61) Pfaendtner, J.; Broadbelt, L. J. Mechanistic Modeling of Lubricant Degradation. 2. The Autoxidation of Decane and Octane. *Ind. Eng. Chem. Res.* **2008**, *47*, 2897–2904.
- (62) Dupin, M.; Reynaud, P.; Jarošík, V.; Baker, R.; Brunel, S.; Eyre, D.; Pergl, J.; Makowski, D. Effects of the Training Dataset Characteristics on the Performance of Nine Species Distribution Models: Application to *Diabrotica Virgifera Virgifera*. *PLoS One* **2011**, *6*, No. e20957.
- (63) Gong, J.; Liu, J.-Y.; Sun, X.-W.; Zheng, B.; Nie, S.-D. Computer-Aided Diagnosis of Lung Cancer: The Effect of Training Data Sets on Classification Accuracy of Lung Nodules. *Phys. Med. Biol.* **2018**, *63*, No. 035036.
- (64) Ding, Y.; Huang, B.; Wu, C.; He, Q.; Lu, K. Kinetic Model and Parameters Study of Lignocellulosic Biomass Oxidative Pyrolysis. *Energy* **2019**, *181*, 11–17.
- (65) Yu, J.; Gou, X.; Yu, J. Comprehensive Chemical Kinetic Model of 2,6,10-Trimethyl Dodecane. *Energy Fuels* **2020**, *34*, 2366–2375.
- (66) Curran, H. J.; Gaffuri, P.; Pitz, W. J.; Westbrook, C. K. A Comprehensive Modeling Study of N-Heptane Oxidation. *Combust. Flame* **1998**, *114*, 149–177.
- (67) Curran, H. J.; Gaffuri, P.; Pitz, W. J.; Westbrook, C. K. A Comprehensive Modeling Study of Iso-Octane Oxidation. *Combust. Flame* **2002**, *129*, 253–280.
- (68) Sarathy, S. M.; Westbrook, C. K.; Mehl, M.; Pitz, W. J.; Togbe, C.; Dagaut, P.; Wang, H.; Oehlschlaeger, M. A.; Niemann, U.; Seshadri, K.; Veloo, P. S.; Ji, C.; Egolfopoulos, F. N.; Lu, T. Comprehensive Chemical Kinetic Modeling of the Oxidation of 2-Methylalkanes from C7 to C20. *Combust. Flame* **2011**, *158*, 2338–2357.