Evolutionary Optimization of the Reduced Gas-phase Isoprene Oxidation Mechanism

Arijit Chakraborty[a,*], Forwood Cloud Wiser[a,*], Siddhartha Sen[b], V. Faye McNeill[a,c,**], Venkat Venkatasubramanian[a,**]

[a] *Department of Chemical Engineering, Columbia University, New York, NY 10027, USA*
[b] *Microsoft Research, New York, NY 10012, USA*
[c] *Department of Earth and Environmental Sciences, Columbia University, New York, NY 10027, USA*

Abstract

Atmospheric chemistry is highly complex, and significant reductions in the size of the chemical mechanism are required to simulate the atmosphere. One of the bottlenecks in creating reduced models is identifying optimal numerical parameters. This process has been difficult to automate, and often relies on manual testing. In this work, we present the application of particle swarm optimization (PSO) towards optimizing the stoichiometric coefficients and rate constants of a reduced isoprene atmospheric oxidation mechanism. Using PSO, we are able to achieve up to 27% improvement in our accuracy metric when compared to a manually tuned reduced mechanism, leading to a significantly optimized final mechanism. This work demonstrates PSO as a promising and thus far underutilized tool for atmospheric chemical mechanism development.

Keywords: Evolutionary optimization, mechanism reduction, mathematical optimization, stoichiometric coefficients, rate parameters, derivative-free optimization

1. Introduction

Model reduction is a common strategy for modeling complex systems that are computationally constrained. Atmospheric isoprene chemistry is one of such systems, where the full extent of the known isoprene chemistry is far larger than can be implemented in 3-dimensional chemical transport models of the atmosphere [1]. These models are used to forecast air quality and climate modeling, to which isoprene is a major contributor. In this article, we present the use of particle swarm optimization (PSO) to optimize stoichiometric coefficients for the recently published AMORE-Isoprene mechanisms v1.1 and v1.2 [1], referred to as AMORE v1.1 and AMORE v1.2 respectively throughout the rest of the text. The AMORE v1.1 and AMORE v1.2 mechanisms are reduced isoprene mechanisms developed from the Caltech isoprene mechanism [2] using

[*] Authors contributed equally
[**] Corresponding author

Email addresses: ac4758@columbia.edu (Arijit Chakraborty), fcw2110@columbia.edu (Forwood Cloud Wiser), sidsen@microsoft.com (Siddhartha Sen), vfm2103@columbia.edu (V. Faye McNeill), venkat@columbia.edu (Venkat Venkatasubramanian)

a mechanism reduction algorithm. The AMORE-Isoprene mechanisms were created using a graph-theory-based algorithm that measures the sensitivity of the full mechanism to a wide range of input conditions and creates a set of reduced mechanistic pathways that have output similar to the full mechanism. This algorithm was motivated by the need to create highly reduced volatile organic compound (VOC) oxidation mechanisms for use in computationally expensive 3D chemical transport models, which are used to model atmospheric aerosol formation, and air quality.

Optimization of stoichiometric coefficients and rate constants for chemical reaction mechanisms is not trivial, and represents a substantial bottleneck in the generation of accurate reduced mechanisms. The reduced model is tasked with accurately representing the full chemistry in terms of the consumption and production of several priority species over a wide range of atmospheric conditions. Measurement of the accuracy of a single reduced mechanism requires simulations of the mechanism under multiple conditions, which leads to a high computational cost to measure the objective function that is being optimized. Additionally, mechanism parameters are highly coupled, and changes in one parameter often impact the optimal value for many other parameters. This means that parameters must be optimized simultaneously and that there are many potential local minima in the objective function. Although reduced mechanisms are considerably smaller than the full mechanisms on which they are based, they still contain a large number of parameters. For example, the AMORE v1.2 mechanism contains 107 stoichiometric coefficients, and 22 rate constants. The high number of coupled parameters to optimize, combined with the relatively slow objective function evaluation time, makes this a challenging optimization problem.

The remainder of this paper is organized as follows: in Section 2, we discuss the problem of chemical reaction modeling, and present a brief overview of the methods used to optimize reduced chemical mechanisms. In Section 3, we outline the details of the particle swarm optimization algorithm, and how it has been adapted to our problem of optimizing stoichiometric parameters for reduced chemical reaction mechanisms. This is followed by presenting the results in Section 4 for the reaction mechanisms under study, atmospheric gas-phase isoprene oxidation, for both variants, namely AMORE v1.1 and AMORE v1.2. Finally, in Section 5 we conclude and summarize our work presented in this article.

2. Background

2.1. Reaction mechanism modeling

Atmospheric chemical modeling is used for predictions and source apportionment of pollutants and particulate matter (PM), as well as being critical for accurate climate modeling [3]. Accurate atmospheric chemical modeling relies on compact and high-quality chemical mechanisms for a range of atmospheric species. VOCs have particularly complex chemistry, and many high-fidelity mechanisms have been developed for such compounds, which are accurate but far too large to be incorporated into atmospheric chemical models [2]. Thus, reduced chemical mechanisms are used to model the VOC chemistry, although some accuracy is lost. Where both complex and reduced mechanisms exist for a given compound, there is the potential to optimize reduced VOC mechanisms against the complex mechanism baseline using representative simulations, which are less computationally costly. However, few tools have been developed to do these optimizations up to this point.

Isoprene was chosen because it is a major component of atmospheric VOC's [4]; it influences tropospheric oxidant levels [5]; it contributes significantly to secondary organic aerosol [6, 7, 8, 9, 10], ozone [11, 12], and formaldehyde [13] which are key factors in air quality; and due to the existence of recently published complex reference [2] and reduced [1] isoprene mechanisms.

2.2. Particle swarm optimization

Particle Swarm Optimization (PSO) is a useful technique that has been deployed in a wide range of applications because of the versatility of the approach for challenging optimization processes. These applications include chemical mechanism analysis [14, 15], parameter estimation [16], dynamic optimization [17, 18, 19], forecasting [20], data clustering [21], training feedforward neural networks [22], robotics [23], smart grid design [24], astronomy [25], manufacturing [26], and additional applications [27]. Within the field of atmospheric chemistry, PSO algorithms have been used for various problems, including parameter optimization for custom instruments [28], identifying atmospheric gas species sources [29, 30, 31], predicting concentrations of select species or pm [32, 33], and estimating particle size distributions [34].

A major benefit of using PSO is that we can choose to impose first-principles-based constraints on the optimization, which include bounds and heuristics for the optimization variables. This is an avenue for the inclusion of domain knowledge in the modeling framework, resulting in a hybrid artificial intelligence (AI) approach[35]. PSO belongs to the class of evolutionary algorithms, which is inspired by the process of evolution as observed in nature. These have had success in domains such as model discovery[36, 37], process systems engineering[38, 39], inverse design[40], materials design[41], and many others[42].

Inspired by the movement of a flock of birds, PSO attempts to model the collective intelligence of particles (or agents) toward the optimization of a global objective while adhering to local rules. It relies on a combination of global and local search by weighting their respective deviations, such that it is able to sufficiently explore the search space of objective variables while honing in on well-performing spaces that result in the optimization of the objective function. Its strength, which enables its applicability to a myriad of domains, is due to having a limited set of tunable parameters, and relatively simple update rules as one proceeds from one iteration to the next. As an evolutionary algorithm, we must point out that one of the drawbacks is that the algorithm does not ensure that the optimal value obtained after the prespecified iterations is the global optimum. Accordingly, we must proceed with additional runs and save the best-performing optimal value(s).

3. Methods

We employ a derivative-free optimization to optimize the stoichiometric coefficients of the reduced chemical mechanisms (AMORE v1.1 and AMORE v1.2). The reason for a derivative-free evolutionary optimization approach is two-fold. Firstly, by virtue of the problem formulation, there is no unique mathematical function that can accurately and reliably map the multiple coefficients (stoichiometric and/or rate parameters) to a continuous function, for every discrete possibility of reactant(s) and product(s). Accordingly, it is not possible to evaluate a gradient of the same. Secondly, the use of an evolutionary optimization scheme allows exploration of the huge parameter space, which is often a shortcoming of gradient-based approaches. As mentioned in Section 1, there are more than 100 parameters for a reduced chemical mechanism. Thus, optimization would need to traverse a combinatorially large space to obtain the best-performing combination of optimization variables.

The evolutionary optimization strategy employed in this article is particle swarm optimization (PSO)[43]. It belongs to a class of *nature-inspired* computing techniques[44] for optimization, termed swarm intelligence[45]. PSO is well-suited to the problem discussed in this article because the search space is high-dimensional (equal to the number of free-roaming stoichiometric coefficients and rate constants), there are multiple local minima in the objective function, and the computational cost to measure the objective function on an individual mechanism is high. This favors an approach, such as PSO, that efficiently explores the search space, is derivative-free, and requires a low number of objective function evaluations. We note that PSO is an exemplar of an evolutionary optimization algorithm that is simple and particularly well-suited to our scenario, but it is not the only evolutionary algorithm that could potentially be applied. In the subsequent subsections 3.1 and 3.2, we discuss the PSO algorithm and the objective function used in this study, respectively.

3.1. Particle swarm optimization

Consider an objective function $f(x) : \mathbb{R}^n \to \mathbb{R}$ that we wish to optimize. For our current problem, we minimize the difference between the concentration of species predicted by our reduced mechanism, and that predicted by the full mechanism. The objective of PSO is to minimize this difference, which it does by changing the values of the stoichiometric parameters of the reduced mechanism such that optimal parameters are obtained. These stoichiometric parameters are the optimization variables in this problem.

At the start of the algorithm, several sets of random optimization variables are generated. These variables can be thought of as *particles* in a space of N dimensions, where each instance is the location of the particle.

Thus, the goal is to find the optimum position of the particles that minimizes the value of the objective function $f(x)$. Let $\overrightarrow{g}(t)$ denote the best position the algorithm has encountered in iteration t, and $\overrightarrow{p}$ denote the best position the algorithm has encountered since the start of the algorithm. Let $\overrightarrow{x}_i(t)$ denote the position of the particle i during iteration t. This position is updated when the particle *moves* to a new position with some velocity $v_i(t)$. These positions are updated based on update rules as follows:

$$v_i(t+1) = \chi * v_i(t) + \phi_1 * \omega_1 * (\overrightarrow{p} - \overrightarrow{x}_i(t)) + \phi_2 * \omega_2 * (\overrightarrow{g}(t) - \overrightarrow{x}_i(t)) \tag{1}$$

$$\overrightarrow{x}_i(t+1) = \overrightarrow{x}_i(t) + v_i(t+1) \tag{2}$$

Here, ω_1 and ω_2 are random numbers uniformly sampled between 0 and 1. These incorporate stochasticity into the calculation of velocity of the particle. ω_1 and ω_2 are constant parameters that weigh the emphasis given to deviation from the best globally and locally performing particles in the swarm respectively. χ is termed the inertia weight, which is a measure of the contribution of the previous velocity of a particle to its current velocity[46]. Based on an agent's new velocity, its position is updated. Together, these terms determine the balance between exploration (global search) and exploitation (local search) in PSO. This is repeated for the prespecified number of iterations until we obtain the best-performing particles. The algorithm ensures that the best-performing particle is at least at par with the optimum in a previous iteration, but not worse, unlike gradient descent which can overshoot depending on the learning rate. This is unlike gradient-based approaches, where due to an incorrect choice of the learning rate, the search for the optimum value(s) across the loss landscape leads to overshooting and/or divergence. Due to PSO algorithm's inherent stochastic nature, it is recommended to run the algorithm for a few runs, as the optimum obtained after the prespecified number of iterations can vary. This also mitigates the risk of getting stuck in a local optimum (here, minimum) of the objective function.

The progression of the PSO on a sample reduced mechanism from one iteration to the next is depicted in Figure 1. Since we use MATLAB[47] for the problem discussed in this article, we refer the reader to the implementation of PSO Global Optimization Toolbox[48], which includes modifications from Mezura-Montes and Coello Coello[49], and Pedersen[50].

3.2. Objective function

In our problem, the objective function is an aggregate measure of the fidelity of the net production rates of priority species as obtained from the reduced mechanism, compared to those obtained from the full mechanism. This is measured under six different conditions shown in Table 1, pertaining to isoprene-relevant

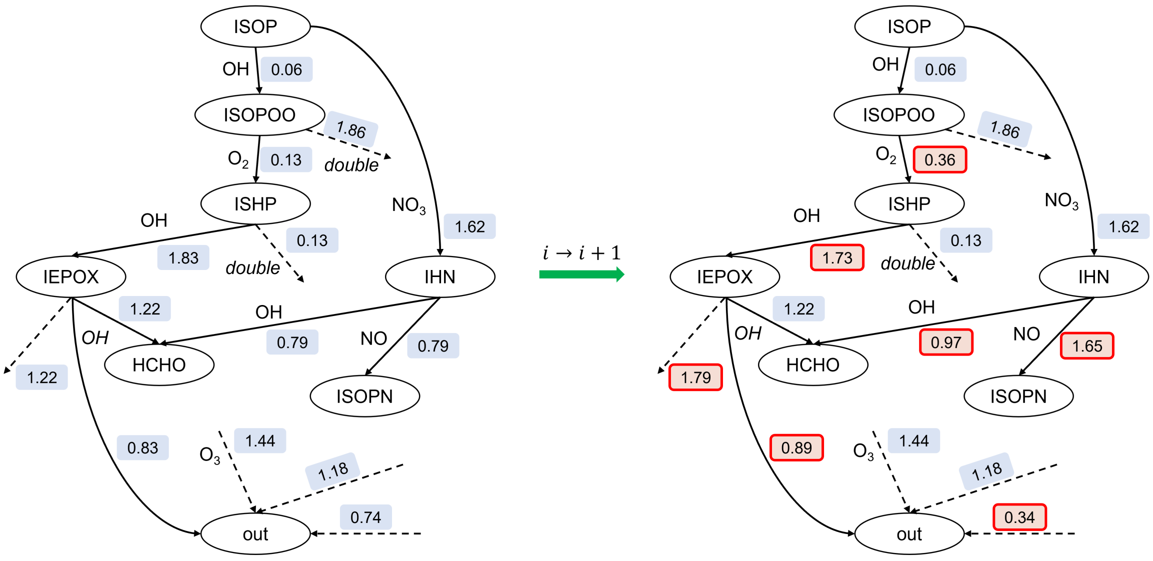

Figure 1: The change in optimization variables – here, stoichiometric coefficients – as PSO proceeds from one iteration to the next. The red-colored boxes in the updated mechanism on the right depict the stoichiometric coefficients with changed values. The node attributes such as *double*, O_3, and others refer to the type of reaction, which remains unchanged throughout the optimization.

conditions that occur in the atmosphere. Thus, we choose to minimize this objective function, as a lower objective function value quantitatively corresponds to a more accurate condensed representation of the full mechanism.

The goal of the objective function is to guide the optimization towards an accurate reduced mechanism that behaves similarly to the original full mechanism. Ultimately, a highly accurate reduced mechanism can be incorporated into a three-dimensional transport model for accurate air quality simulations. However, these models are highly computationally expensive, taking on the order of hours to days to simulate on supercomputers. This means that the final use case is not applicable for optimization with multiple runs, where a rapid evaluation of the objective function is necessary. Therefore, we used a standard method for testing chemical mechanisms, a box model, which does not have a spatial component, greatly reducing computational costs. Box model simulations can be run under a set of invariant conditions (temperature, pressure, solar intensity, and concentrations of reactive background species), which are chosen based on frequently encountered atmospheric conditions. The box model used in this work is the F0AM box model [51], which runs in MATLAB. A 24-hour simulation of a candidate mechanism under one set of conditions takes approximately 0.8 seconds to run. We used a sample of representative conditions meant to capture the variety seen in the atmosphere. In general, greater or fewer input conditions can be selected, inducing a trade-off between computational cost and atmospheric representation. This trade-off is also influenced by the variety of situations in which the mechanism being tested is relevant. In the case of the isoprene mechanism, we chose six different input conditions meant to select the most relevant conditions for isoprene. Table 1

Table 1: Six different run conditions used to evaluate mechanisms. All species values have units of ppb. Photolysis is a unitless constant.

Sample	Run Description	ISOP	OH	HO_2	NO	O_3	NO_3	RO_2	Photolysis
1	High OH	5	0.0002	0.007	0.01	0	0	0.001	1
2	High OH and NO	5	0.0002	0.007	0.2	0	0	0.001	0
3	High O_3	2	0.00001	0.007	0.01	100	0	0.001	1
4	High NO_3	1	0.00001	0.007	0.1	0	0.0002	0.001	1
5	High NO_3 & no $h\nu$	1	0.00001	0.007	0.1	0	0.0002	0.001	0
6	High Isoprene	10	0.0002	0.007	0.02	0	0	0.001	1

lists these conditions. The set of conditions is provided as an input to the objective function evaluation, and all mechanisms are evaluated on all conditions. Although not addressed here, optimally selecting the input conditions is an orthogonal problem to pursue in future work.

The isoprene mechanism influences several important atmospheric species, including OH, HO_2, NO, NO_2, ozone (O_3), formaldehyde (HCHO), and isoprene epoxy-diol (IEPOX), lumped isoprene nitrates (ISOPN), glyoxal (GLY), methylglyoxal (MGLY), methyl vinyl ketone (MVK), and methacrolein (MACR). The function includes individual performance metrics for each of the priority species involved in the mechanism, which are given an importance weighting based on the environmental context. In order to take into consideration the performance of the mechanism across multiple species and conditions, the objective function consists of a weighted average of individual species-run performance metrics. A species-run is defined as a simulation of an individual species under one set of input conditions.

The ultimate performance goal of the reduced mechanism is to accurately match the concentration of the priority species in the full mechanism. The rate of production and consumption are the two forces that influence the overall concentration of the priority species. The isoprene mechanism influences primarily the production rate of several priority organic species and also the production and consumption rate of some reactive background species. A useful species-run metric is bounded, so that averages can be taken without being skewed by significantly higher or lower values. The production and consumption rate of the priority species in the isoprene mechanism varies over time, with an increase in oxidation of isoprene and its products. The species-run metric captures this time dependence by integrating the difference in production and consumption rates of the target species between the test and reference mechanism over the entire run time. It must be noted that the reference mechanism was run on the same box model. The sum of the reference and test values is used as the denominator so that the quantity is normalized to be less than or equal to one. The following equations give the metric used for the individual species run, which was averaged to create the objective function:

$$P_{x,s}^{T} = \int_{t_0}^{t_f} p_{x,s}^{T}(t)dt \tag{3}$$

$$P_{x,s}^{R} = \int_{t_0}^{t_f} p_{x,s}^{R}(t)dt \tag{4}$$

$$C_{x,s}^{T} = \int_{t_0}^{t_f} c_{x,s}^{T}(t)dt \tag{5}$$

$$C_{x,s}^{R} = \int_{t_0}^{t_f} c_{x,s}^{R}(t)dt \tag{6}$$

$$f_{x,s}(T,R) = \frac{abs[(P_{x,s}^{T} - \alpha C_{x,s}^{T}) - (P_{x,s}^{R} - \alpha C_{x,s}^{R})]}{abs[(P_{x,s}^{T} - \alpha C_{x,s}^{T}) + (P_{x,s}^{R} - \alpha C_{x,s}^{R})]} \tag{7}$$

Here, x represents a set of input conditions, s represents the priority species being measured, T denotes that the test mechanism is being measured, R denotes that the reference mechanism is being measured, $p_{x,s}^{T}(t)$ represents the rate of production of species s with input conditions x using mechanism T, $c_{x,s}^{T}(t)$ represents the rate of consumption of the same, α_s is a binary variable which denotes whether or not consumption should be taken into account for species s, $C_{x,s}^{T}$ and $P_{x,s}^{T}$ represent the total net consumption and production of species s with input conditions x for mechanism T over the total run time from t_0 to t_f respectively, and $f_{x,s}(T,R)$ represents the species-run performance metric. The performance metric ranges from 0 to 1, where 0 represents perfect alignment with the entire mechanism, and 1 represents an infinite deviation from the reference mechanism. Only test mechanisms that represent and match the production and consumption rate of the target species throughout the entire run-time will have performance metrics that are close to 0. Equation 8 shows the overall objective function used for a test mechanism.

$$F(T,R) = \sum_{x=0}^{X} \sum_{s=0}^{S} \omega_s f_{x,s}(T,R) \tag{8}$$

Here, $F(T,R)$ is the objective function for a mechanism T compared to the reference mechanism R, X represents the set of all test conditions, S represents all the priority species being measured, ω_s represents the weighting assigned to a given species, and $f_{x,s}(T,R)$ is given in equation 7. By virtue of the problem formulation, we can explore a few orders of magnitude of the acceptable rate parameter coefficients, and similarly, for stoichiometric coefficients, we can search within a user-defined range. Here, the rate parameters were allowed to perturb within 2 orders of magnitude of the previously user-defined default values, which served as a reasonable starting point for the algorithm. The stoichiometric coefficients of the products were restricted to be within 0.01 to 2. The reactant stoichiometric coefficients were held constant. These stoichiometric coefficients and rate parameter values are the optimization variables used in PSO. We first optimize only the stoichiometric coefficients while keeping the rate parameters constant, and obtain their results. Separately, we optimized the stoichiometric coefficients and rate parameters simultaneously. This was done in order to investigate the effect of including rate parameters on optimization results.

In the next section, we present the results of optimizing the AMORE v1.1 and AMORE v1.2 reduced mechanisms, using both: only stoichiometric parameter optimization, and stoichiometric and rate parameter optimization. The results of the same are compared to the concentration plots obtained from the AMORE v1.1, AMORE v1.2, and the Caltech Isoprene mechanism designed by human experts [1][2].

4. Results and Discussion

We conducted several runs of the PSO algorithm on the AMORE v1.2 mechanism. All PSO-optimized mechanisms scored better on the objective function than the AMORE v1.2 baseline mechanism. We ran the optimization using different population sizes, and number of generations. Based on conventional evolutionary optimization terminology, *population* refers to the entire collection of optimization variables. Thus, a population of 50 individuals would have 50 instances of N-dimensional optimization variables, with each set of N-dimensional optimization variables being referred to as an *individual*. *Generation* refers to the iterations of the optimization algorithm.

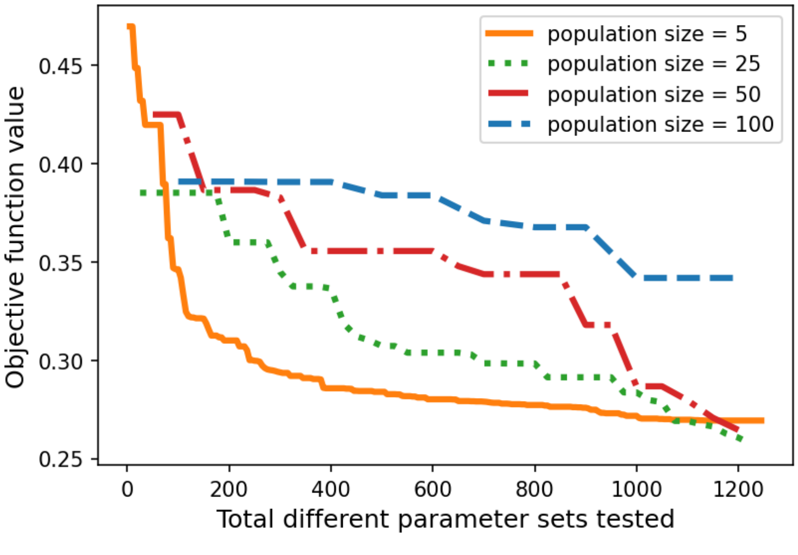

Figure 2: Best objective function score within the population plotted against the number of different parameter sets tested. Plots shown for multiple PSO runs with varying population sizes of 5, 25, 50, and 100.

Figure 2 shows the best individual performance within the current population versus the number of parameter sets tested for several different population sizes. The x-axis scales with the run time, as testing each parameter set takes roughly the same amount of time. The starting fitness for each run is different due to the stochastic nature of the initial particle selection. Although larger populations will have lower starting values on average, there is no guarantee for an individual run. Each run starts at a different number of parameter sets tested, since the starting point represents the best fitness after the first generation of parameter sets has been tested. In all cases, the objective function decreases rapidly at first and more slowly as the optimization goes on. From the data, we can see that, initially, small population sizes are able to descend more rapidly towards a better objective function score, but more quickly reach a plateau where the descent is much more gradual. Larger population sizes tend to show a much lower initial descent that is sustained for a longer period, and eventually tend to reach lower plateau values. This can be explained by the fact that for larger population sizes, each generation requires more parameter sets to be tested, leading to a much slower convergence towards the vicinity of the best particle. However, having more particles leads to more of the search space being explored over longer run times, and thus lower final values. Note that due to run time constraints, we did not find the plateau values for larger population sizes, but the 100 set population size eventually reached a fitness of 0.213 after 5500 parameter sets tested. We The practical implication of this is that the population size should scale with the intended run-time of the algorithm. However, the dataset is not large enough to draw conclusions about the optimal population size and, due to the stochastic nature of the algorithm, the results will vary significantly between runs. It takes approximately 5 seconds to test a single parameter set (on a Dell 2000 MHz Inspiron 15 laptop with 16 GB RAM) and at least 60 generations for a given run to make the bulk of its improvements; thus, a population size should be chosen which will allow for at least 60 generations within the desired run time. However, shorter runs with as few as 25 generations have shown to be useful in reducing the objective function significantly. For our optimizations, we chose a population size of 50, as we were able to accommodate the requisite run time to make that population size worthwhile. After performing several optimizations using the same settings, we found that the final values chosen were very different between runs, indicating that each run tends to find its own local minimum. This is expected, and is attributed to the stochastic nature of PSO. In addition, constraining some parameters leads to a small reduction in variation of other parameters, suggesting that further constraints on the search space will lead the optimization results to be less varied. Despite variation in the final parameter values, runs with the same settings tended to converge to very similar fitness values.

We chose a selection of our best PSO mechanisms for a more detailed analysis. These mechanisms include two optimized variants of both, the AMORE v1.1 isoprene mechanism, and the AMORE v1.2 isoprene mechanism. For both, we first optimize only the stoichiometric coefficients, followed by optimization of stoichiometric coefficients and rate parameters. All optimizations were performed for 100 generations with a population size of 50. Table 2 shows the fitness values for each optimized mechanism, and the mechanism it is optimizing. The optimization of the PSO without rate coefficients was able to improve the AMORE v1.1

Table 2: Tabulated results for the measured fitness values for six reduced isoprene mechanisms under the six different testing conditions. 'Baseline' refers to the unoptimized baseline mechanism as obtained by manual tuning. 'Coefficients' refers to the mechanism optimized only on its stoichiometric coefficients. 'Coefficients + Rates' refers to the mechanism with optimized stoichiometric coefficients, and rate parameters.

Sample	Mechanism	AMORE v1.1			AMORE v1.2		
		Baseline	Coefficients	Coefficients + Rates	Baseline	Coefficients	Coefficients + Rates
1	High OH	0.49	0.31	0.24	0.28	0.19	0.19
2	High OH+NO	0.43	0.37	0.44	0.31	0.25	0.28
3	High O_3	0.30	0.10	0.26	0.29	0.19	0.17
4	High NO_3	0.37	0.32	0.31	0.29	0.20	0.25
5	High NO_3 no hv	0.37	0.38	0.42	0.32	0.28	0.34
6	High Isoprene	0.46	0.31	0.22	0.28	0.17	0.19
	Average	0.40	0.30	0.31	0.29	0.21	0.24
	% improvement	-	**25.7**	**22.0**	-	**27.4**	**19.5**

mechanism by 25.7%, and improve the AMORE v1.2 mechanism by 27.4%. With the rate constants included in optimization, the AMORE v1.1 mechanism was improved by 22.0% and the AMORE v1.2 mechanism was improved by 19.5%. The inclusion of rate coefficients reduced the level of improvement, which suggests that the rate constants in the original mechanism were well-calibrated, and the increased size of the search space outweighed the benefit of having more changeable parameters. The PSO optimization had strong breadth of improvement in all six testing conditions. The two optimizations without rate constants improved the fitness in 5/6 and 6/6 of the testing conditions, and the optimization with rate constants included results in improved fitness on 4/6 and 5/6 of the testing conditions.

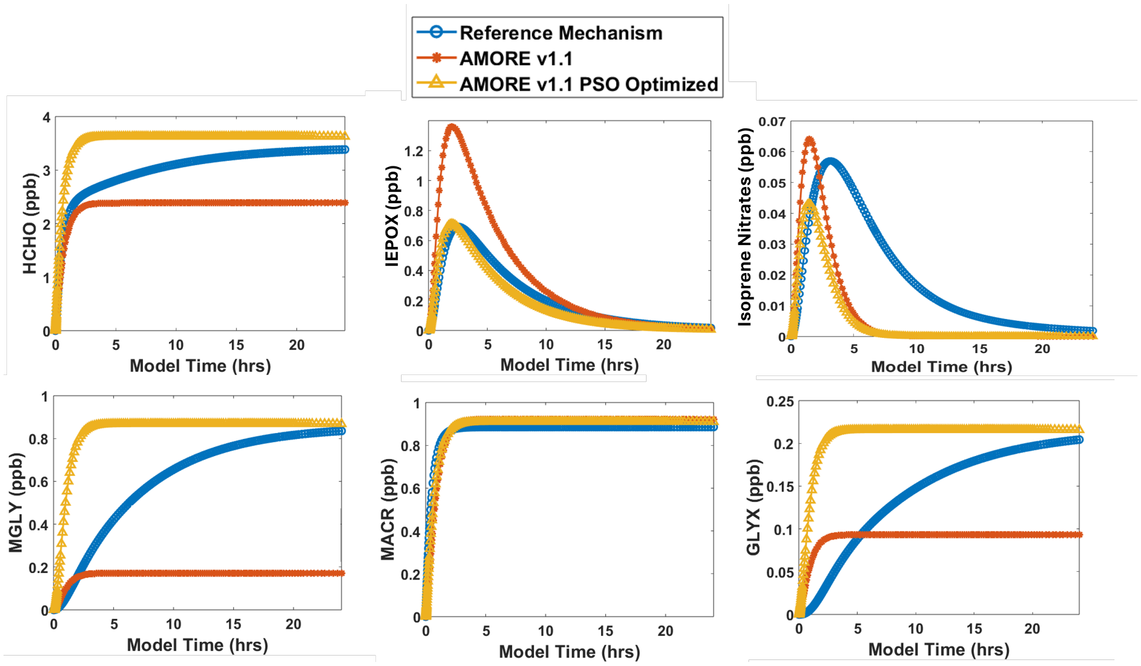

Figure 3: Example concentration plots of the reference mechanism, AMORE v1.1 mechanism, and AMORE v1.1 50x100 PSO-optimized mechanism without rate constant optimization, run for a population size of 50, and 100 generations. The measured fitness values for AMORE v1.1 are: HCHO, 0.17; IEPOX, 0.32; ISOPN, 0.72; MGLY, 0.66; MACR, 0.02; GLY, 0.38. The measured fitness values for AMORE v1.1 50x100 PSO are: HCHO, 0.04; IEPOX, 0.00; ISOPN, 0.78; MGLY, 0.02; MACR, 0.01; GLY, 0.01. The run input condition is the high OH test condition from Table 1.

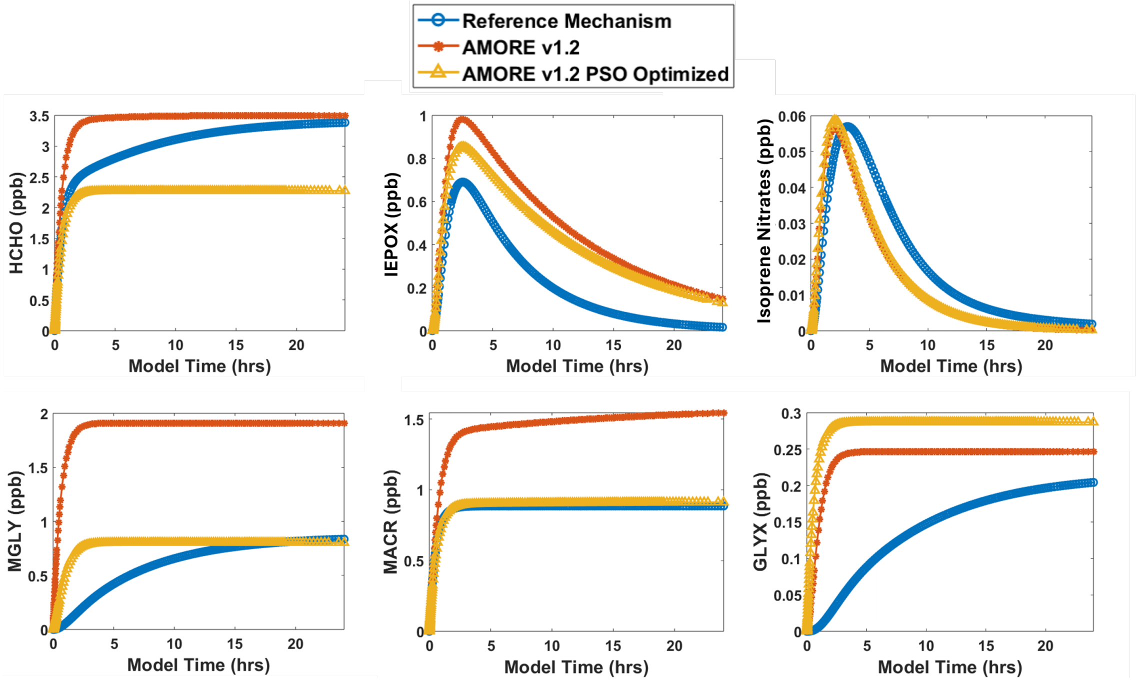

Figure 4: Example concentration plots of the reference mechanism, AMORE v1.2 mechanism, and AMORE v1.2 50x100 PSO-optimized mechanism without rate constant optimization, run for a population size of 50, and 100 generations. The measured fitness values for AMORE v1.2 are: HCHO, 0.02; IEPOX, 0.08; ISOPN, 0.38; MGLY, 0.40; MACR, 0.29; GLY, 0.08. The measured fitness values for AMORE v1.2 50x100 PSO are: HCHO, 0.19; IEPOX, 0.01; ISOPN, 0.37; MGLY, 0.02; MACR, 0.02; GLY, 0.16. The run input condition is the high OH test condition from Table 1.

Figures 3 and 4 show the concentration of a select set of organic species (formaldehyde (HCHO), isoprene epoxy-diol (IEPOX), lumped isoprene nitrates, methylglyoxal (MGLY), methacrolein (MACR), and glyoxal (GLYX)) under high OH conditions for several isoprene mechanisms. Figure 3 shows the AMORE v1.1 mechanism in comparison to the AMORE v1.1 PSO-optimized mechanism (population size of 50, for 100 generations), without rate constant optimization, alongside the full reference isoprene mechanism (here, the Caltech isoprene mechanism). Figure 4 shows the AMORE v1.2 mechanism in comparison to the AMORE v1.2 PSO-optimized mechanism (population size of 50, for 100 generations), without rate constants, alongside the full reference isoprene mechanism (here, the Caltech isoprene mechanism).

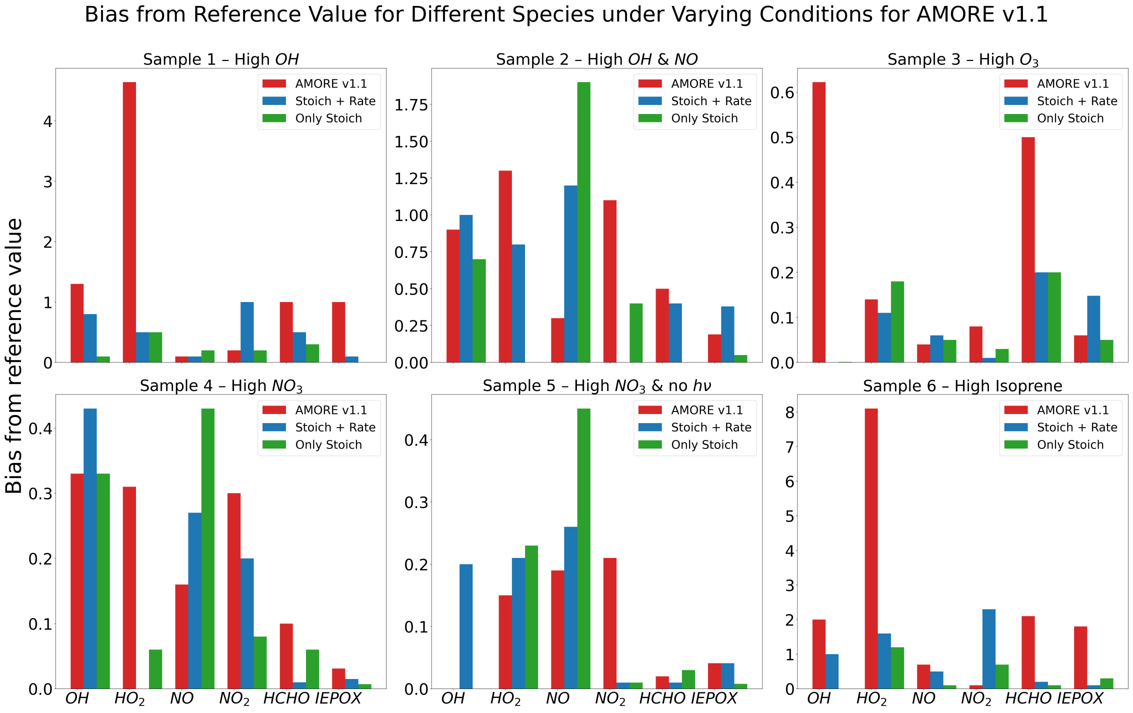

Figure 5: Bias from reference value for the optimized AMORE v1.1 mechanism for 6 different conditions (Table 1), and the 6 species – OH, HO_2, NO, NO_2, $HCHO$, and $IEPOX$. There is considerable decline in the bias for the optimized mechanisms (blue and green), when compared to manually edited AMORE v1.1 mechanism (red).

Figures 5 and 6 compare the deviation from reference value of PSO-optimized mechanisms to the AMORE baseline mechanisms for six of the most important species under the six different testing conditions (specified in Table 1). There is variation in the deviations between species and conditions, but on average, there is a significant reduction in deviations from the AMORE mechanisms to the PSO mechanisms.

5. Conclusion

In this paper, we present an optimization approach for obtaining the optimal parameters of a reduced chemical mechanism, such that the fidelity to the full mechanism is maximized. The approach relies on

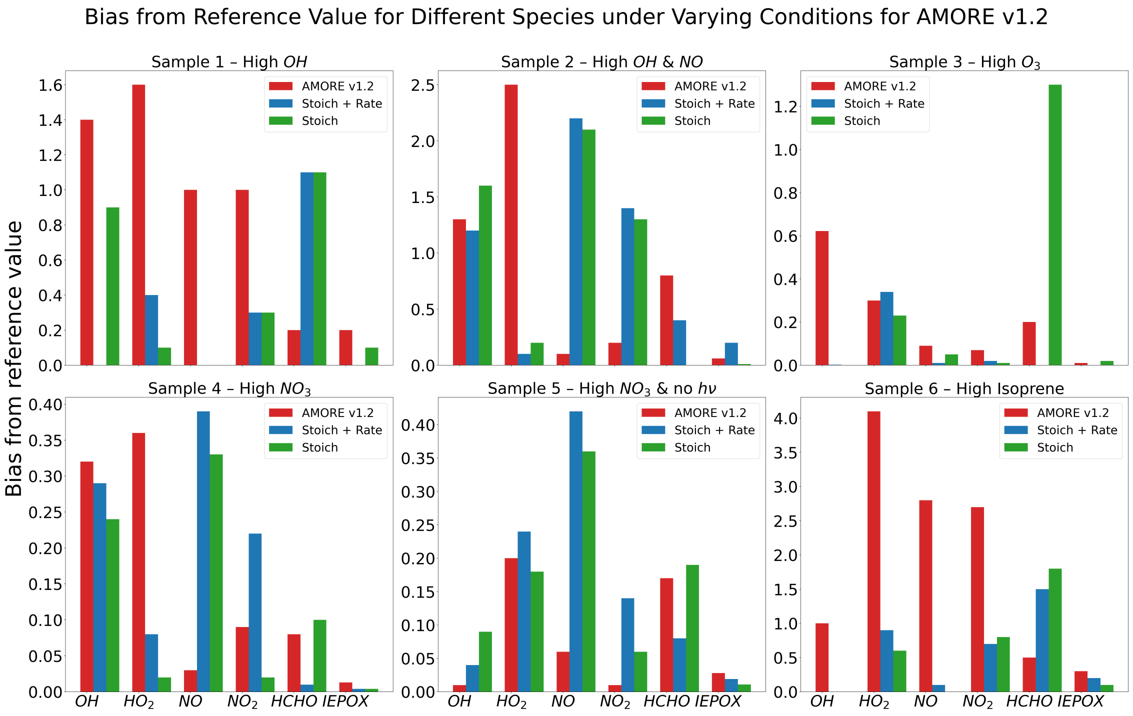

Figure 6: Bias from reference value for the optimized AMORE v1.2 mechanism for 6 different conditions (Table 1, and the 6 species – OH, HO_2, NO, NO_2, $HCHO$, and $IEPOX$. There is considerable decline in the bias for the optimized mechanisms (blue and green), when compared to manually edited AMORE v1.2 mechanism (red).

the popular and effective evolutionary optimization algorithm, particle swarm optimization (PSO). We have discussed the results for optimization of only stoichiometric coefficients, and that of stoichiometric coefficients and rate parameters simultaneously. The latter results in a larger search space due to additional objective variables to be optimized, which PSO is able to handle reasonably well.

The benefits accrued from the optimization of parameters of a reduced mechanism are its increased accuracy when compared to the complete large-scale mechanism. Such an optimized reduced mechanism can be used independently for making predictions of the concentrations of important species in the atmosphere, for a fraction of the computational power in comparison to the full reference mechanism. While the parameters obtained are not globally optimum, the approach yields optimal parameter values for both the reduced mechanisms considered in this article, with an increase of up to 27% in the objective function over the baseline state-of-the-art mechanism.

Acknowledgements

This publication was developed under Assistance Agreement No. 84001301 awarded by the U.S. Environmental Protection Agency to McNeill. The views expressed in this article are those of the authors and do not necessarily represent the views or policies of the U.S. Environmental Protection Agency. EPA does not endorse any products or commercial services mentioned in this publication. VV is grateful for support in part by the NSF EFRI-DCheM 2132142 grant and funding from the Center for the Management of Systemic Risk (CMSR) at Columbia University.

Data Availability

The code for the work presented in this article is made available online: https://github.com/fcw2110/GA-PSO-AMORE (last access: June 5, 2024).

Conflicts of interest

There are no conflicts of interest to declare.

References

[1] F. Wiser, B. K. Place, S. Sen, H. O. Pye, B. Yang, D. M. Westervelt, D. K. Henze, A. M. Fiore, V. F. McNeill, Amore-isoprene v1. 0: A new reduced mechanism for gas-phase isoprene oxidation, Geoscientific Model Development 16 (2023) 1801–1821.

[2] P. O. Wennberg, K. H. Bates, J. D. Crounse, L. G. Dodson, R. C. McVay, L. A. Mertens, T. B. Nguyen, E. Praske, R. H. Schwantes, M. D. Smarte, J. M. St Clair, A. P. Teng, X. Zhang, J. H. Seinfeld, Gas-phase reactions of isoprene and its major oxidation products, Chemical Reviews 118 (2018) 3337–3390. PMID: 29522327.

[3] H. O. T. Pye, B. K. Place, B. N. Murphy, K. M. Seltzer, E. L. D'Ambro, C. Allen, I. R. Piletic, S. Farrell, R. H. Schwantes, M. M. Coggon, E. Saunders, L. Xu, G. Sarwar, W. T. Hutzell, K. M. Foley, G. Pouliot, J. Bash, W. R. Stockwell, Linking gas, particulate, and toxic endpoints to air emissions in the community regional atmospheric chemistry multiphase mechanism (cracmm) version 1.0, Atmospheric Chemistry and Physics Discussions 2022 (2022) 1–88.

[4] A. Guenther, T. Karl, P. Harley, C. Wiedinmyer, P. I. Palmer, C. Geron, Estimates of global terrestrial isoprene emissions using megan (model of emissions of gases and aerosols from nature), Atmospheric Chemistry and Physics 6 (2006) 3181–3210.

[5] D. Butler, T. M. aand Taraborrelli, C. Brühl, H. Fischer, H. Harder, M. Martinez, J. Williams, M. G. Lawrence, J. Lelieveld, Improved simulation of isoprene oxidation chemistry with the echam5/messy chemistry-climate model: lessons from the gabriel airborne field campaign, Atmospheric Chemistry and Physics 8 (2008) 4529–4546.

[6] J. H. Kroll, N. L. Ng, S. M. Murphy, R. C. Flagan, J. H. Seinfeld, Secondary organic aerosol formation from isoprene photooxidation, Environmental Science & Technology 40 (2006) 1869–1877. PMID: 16570610.

[7] D. K. Henze, J. H. Seinfeld, Global secondary organic aerosol from isoprene oxidation, Geophysical Research Letters 33 (2006).

[8] D. K. Farmer, A. Matsunaga, K. S. Docherty, J. D. Surratt, J. H. Seinfeld, P. J. Ziemann, J. L. Jimenez, Response of an aerosol mass spectrometer to organonitrates and organosulfates and implications for atmospheric chemistry, Proceedings of the National Academy of Sciences 107 (2010) 6670–6675.

[9] J. Liu, E. L. D'Ambro, B. H. Lee, F. D. Lopez-Hilfiker, R. A. Zaveri, J. C. Rivera-Rios, F. N. Keutsch, S. Iyer, T. Kurten, Z. Zhang, A. Gold, J. D. Surratt, J. E. Shilling, J. A. Thornton, Efficient isoprene secondary organic aerosol formation from a non-iepox pathway, Environmental Science & Technology 50 (2016) 9872–9880. PMID: 27548285.

[10] T.-M. Fu, D. J. Jacob, C. L. Heald, Aqueous-phase reactive uptake of dicarbonyls as a source of organic aerosol over eastern north america, Atmospheric Environment 43 (2009) 1814–1822.

[11] A. M. Fiore, H. Levy II, D. A. Jaffe, North american isoprene influence on intercontinental ozone pollution, Atmospheric Chemistry and Physics 11 (2011) 1697–1710.

[12] J. J. Guo, A. M. Fiore, L. T. Murray, D. A. Jaffe, J. L. Schnell, C. T. Moore, G. P. Milly, Average versus high surface ozone levels over the continental usa: model bias, background influences, and interannual variability, Atmospheric Chemistry and Physics 18 (2018) 12123–12140.

[13] G. M. Wolfe, J. Kaiser, T. F. Hanisco, F. N. Keutsch, J. A. de Gouw, J. B. Gilman, M. Graus, C. D. Hatch, J. Holloway, L. W. Horowitz, B. H. Lee, B. M. Lerner, F. Lopez-Hilifiker, J. Mao, M. R. Marvin, J. Peischl, I. B. Pollack, J. M. Roberts, T. B. Ryerson, J. A. Thornton, P. R. Veres, C. Warneke, Formaldehyde production from isoprene oxidation across no$_x$ regimes, Atmospheric Chemistry and Physics 16 (2016) 2597–2610.

[14] H. Wang, C. Sun, O. Haidn, A. Aliya, C. Manfletti, N. Slavinskaya, A joint hydrogen and syngas chemical kinetic model optimized by particle swarm optimization, Fuel 332 (2023) 125945.

[15] C. O. Ourique, E. C. Biscaia, J. C. Pinto, The use of particle swarm optimization for dynamical analysis in chemical processes, Computers Chemical Engineering 26 (2002) 1783–1793.

[16] M. Schwaab, E. C. Biscaia, Jr., J. L. Monteiro, J. C. Pinto, Nonlinear parameter estimation through particle swarm optimization, Chemical Engineering Science 63 (2008) 1542–1552.

[17] Y. Zhou, C. Zhao, X. Liu, An iteratively adaptive particle swarm optimization approach for solving chemical dynamic optimization problems, CIESC J. 65 (2014) 1296–1302.

[18] C. O. Ourique, E. C. Biscaia, J. C. Pinto, The use of particle swarm optimization for dynamical analysis in chemical processes, Computers Chemical Engineering 26 (2002) 1783–1793.

[19] V. Mann, A. Sivaram, L. Das, V. Venkatasubramanian, Robust and efficient swarm communication topologies for hostile environments, Swarm and Evolutionary Computation 62 (2021).

[20] Y. Wang, Y. Ni, N. Li, S. Lu, S. Zhang, Z. Feng, J. Wang, A method based on improved ant lion optimization and support vector regression for remaining useful life estimation of lithium-ion batteries, Energy Science & Engineering 7 (2019) 2797–2813.

[21] S. Alam, G. Dobbie, Y. S. Koh, P. Riddle, S. Ur Rehman, Research on particle swarm optimization based clustering: A systematic review of literature and techniques, Swarm and Evolutionary Computation 17 (2014) 1–13.

[22] J.-R. Zhang, J. Zhang, T.-M. Lok, M. R. Lyu, A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training, Applied Mathematics and Computation 185 (2007) 1026–1037. Special Issue on Intelligent Computing Theory and Methodology.

[23] E. Camci, D. R. Kripalani, L. Ma, E. Kayacan, M. A. Khanesar, An aerial robot for rice farm quality inspection with type-2 fuzzy neural networks tuned by particle swarm optimization-sliding mode control hybrid algorithm, Swarm and Evolutionary Computation 41 (2018) 1–8.

[24] A. El-Zonkoly, Optimal placement of multi-distributed generation units including different load models using particle swarm optimization, Swarm and Evolutionary Computation 1 (2011) 50–59.

[25] N. Jin, Y. Rahmat-Samii, Analysis and particle swarm optimization of correlator antenna arrays for radio astronomy applications, IEEE Transactions on Antennas and Propagation 56 (2008) 1269–1279.

[26] T. Navalertporn, N. V. Afzulpurkar, Optimization of tile manufacturing process using particle swarm optimization, Swarm and Evolutionary Computation 1 (2011) 97–109.

[27] M. Pluhacek, R. Senkerik, A. Viktorin, T. Kadavy, I. Zelinka, A review of real-world applications of particle swarm optimization algorithm, pp. 115–122.

[28] H. Tong, A. M. Arangio, P. S. J. Lakey, T. Berkemeier, F. Liu, C. J. Kampf, W. H. Brune, U. Pöschl, M. Shiraiwa, Hydroxyl radicals from secondary organic aerosol decomposition in water, Atmospheric Chemistry and Physics 16 (2016) 1761–1771.

[29] D. Ma, S. Wang, Z. Zhang, Hybrid algorithm of minimum relative entropy-particle swarm optimization with adjustment parameters for gas source term identification in atmosphere, Atmospheric Environment 94 (2014) 637–646.

[30] J. Wang, R. Zhang, Y. Yan, X. Dong, J. M. Li, Locating hazardous gas leaks in the atmosphere via modified genetic, mcmc and particle swarm optimization algorithms, Atmospheric Environment 157 (2017) 27–37.

[31] D. Ma, W. Tan, Q. Wang, Z. Zhang, J. Gao, Q. Zeng, X. Wang, F. Xia, X. Shi, Application and improvement of swarm intelligence optimization algorithm in gas emission source identification in atmosphere, Journal of Loss Prevention in the Process Industries 56 (2018) 262–271.

[32] J. Zhang, F. Tittel, L. Gong, R. Lewicki, R. Griffin, W. Jiang, B. Jiang, M. Li, Support vector machine modeling using particle swarm optimization approach for the retrieval of atmospheric ammonia concentrations, Environmental Modeling Assessment 21 (2016).

[33] G. N. Kouziokas, Svm kernel based on particle swarm optimized vector and bayesian optimized svm in atmospheric particulate matter forecasting, Applied Soft Computing 93 (2020) 106410.

[34] Y. Yuan, H.-L. Yi, Y. Shuai, F.-Q. Wang, H.-P. Tan, Inverse problem for particle size distributions of atmospheric aerosols using stochastic particle swarm optimization, Journal of Quantitative Spectroscopy and Radiative Transfer 111 (2010) 2106–2114.

[35] A. Chakraborty, S. Serneels, H. Claussen, V. Venkatasubramanian, Hybrid ai models in chemical engineering–a purpose-driven perspective, Computer Aided Chemical Engineering 51 (2022) 1507–1512.

[36] A. Chakraborty, A. Sivaram, L. Samavedham, V. Venkatasubramanian, Mechanism discovery and model identification using genetic feature extraction and statistical testing, Computers & Chemical Engineering 140 (2020) 106900.

[37] A. Chakraborty, A. Sivaram, V. Venkatasubramanian, Ai-darwin: A first principles-based model discovery engine using machine learning, Computers & Chemical Engineering 154 (2021) 107470.

[38] P. Jul-Rasmussen, A. Chakraborty, V. Venkatasubramanian, X. Liang, J. K. Huusom, Identifying first-principles models for bubble column aeration using machine learning, in: Computer Aided Chemical Engineering, volume 52, Elsevier, 2023, pp. 1089–1094.

[39] P. Jul-Rasmussen, A. Chakraborty, V. Venkatasubramanian, X. Liang, J. K. Huusom, Hybrid ai modeling techniques for pilot scale bubble column aeration: A comparative study, Computers & Chemical Engineering (2024) 108655.

[40] V. Venkatasubramanian, K. Chan, J. M. Caruthers, Evolutionary design of molecules with desired properties using the genetic algorithm, Journal of Chemical Information and Computer Sciences 35 (1995) 188–195.

[41] B. Srinivasan, T. Vo, Y. Zhang, O. Gang, S. Kumar, V. Venkatasubramanian, Designing dna-grafted particles that self-assemble into desired crystalline structures using the genetic algorithm, Proceedings of the National Academy of Sciences 110 (2013) 18431–18435.

[42] J. Fang, W. Liu, L. Chen, S. Lauria, A. Miron, X. Liu, A survey of algorithms, applications and trends for particle swarm optimization, International Journal of Network Dynamics and Intelligence (2023) 24–50.

[43] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95-international conference on neural networks, volume 4, IEEE, pp. 1942–1948.

[44] S. Patnaik, X.-S. Yang, K. Nakamatsu, Nature-inspired computing and optimization, volume 10, Springer, 2017.

[45] I. Fister Jr, X.-S. Yang, I. Fister, J. Brest, D. Fister, A brief review of nature-inspired algorithms for optimization, arXiv preprint arXiv:1307.4186 (2013).

[46] J. C. Bansal, P. Singh, M. Saraswat, A. Verma, S. S. Jadon, A. Abraham, Inertia weight strategies in particle swarm optimization, in: 2011 Third world congress on nature and biologically inspired computing, IEEE, pp. 633–640.

[47] T. M. Inc., Matlab version: 23.2.0.2391609 (r2023b), 2023.

[48] T. M. Inc., Optimization toolbox version: 23.2 (r2023b), 2023.

[49] E. Mezura-Montes, C. A. C. Coello, Constraint-handling in nature-inspired numerical optimization: past, present and future, Swarm and Evolutionary Computation 1 (2011) 173–194.

[50] M. E. H. Pedersen, Good parameters for particle swarm optimization, Hvass Lab., Copenhagen, Denmark, Tech. Rep. HL1001 (2010) 1551–3203.

[51] G. M. Wolfe, M. R. Marvin, S. J. Roberts, K. R. Travis, J. Liao, The framework for 0-d atmospheric modeling (f0am) v3.1, Geoscientific Model Development 9 (2016) 3309–3319.