On the potential of Particle Swarm algorithm for the optimization of detailed kinetic mechanisms. Comparison with Genetic Algorithm.

Elissa El Rassy,[†] Aurélie Delaroque,[‡] Patrick Sambou,[‡] Harish Kumar Chakravarty,[‡] and Alexis Matynia[*,‡]

†*Laboratoire de Thermique et Energie de Nantes, UMR 6607, Rue Christian Pauc, BP 50609, 44306 Nantes cedex 03*

‡*Sorbonne Université, Centre National de la Recherche Scientifique, UMR 7190, Institut Jean Le Rond ∂'Alembert, F-75005 Paris, France*

E-mail: alexis.matynia@sorbonne-universite.fr

Abstract

This work investigates the potential of the particle swarm algorithm for the optimization of detailed kinetic mechanisms. To that end, empirical analysis has been conducted to evaluate the efficiency of this algorithm in comparison with the genetic algorithm. Both algorithms are built on evolutionary processes according to which a randomly defined population will evolve, over the iterations, towards an optimal solution. The genetic algorithm is driven by crossover and mutation operators and by a selection method. The PSO approach is based on the experience of each individual and on the group experience to control the direction of its evolution. The success of the application of an algorithm can be sensitive to the choice of operators and the relative importance attributed to them. Therefore, to make the comparison as rigorous as possible, about a dozen strategies were proposed for each algorithm and the performances were evaluated. A degraded version of the GRI-Mech 3.0 mechanism (i.e. containing some of the kinetic constants randomly modified) was generated and then optimized by the two evolutionary algorithms to recover the predictive character of the original mechanism. The results show that, for the majority of the proposed strategies, PSO is more efficient than the GA, whereas the latter is generally much more used for the optimization of detailed kinetic mechanisms.

1 Introduction

The chemical processes that take place during combustion can be described using detailed kinetic mechanisms. Depending on the size of the fuel involved and the degree of precision desired, these mechanisms can be composed of a few very simplified chemical steps or, on the contrary, of several thousand elementary reactions.[1] The reaction rates can be determined experimentally,[2] by means of theoretical approaches[3] or by analogy with reactions of the same type (H-abstration / beta-scission, etc.)[4] Nevertheless, the complexity of the phenomenology of a chemical reaction implies that the uncertainties on the rate constants generally remain relatively large.[5] Similarly, the development of models by combining sets of reactions, even with a high degree of confidence in the kinetic constants, does not always guarantee good predictivity.[6]

The optimization of a kinetic mechanism implies the improvement of its predictability by the variation of the reaction rate parameters within their uncertainty limits. A distinctive feature inherent to the optimization of detailed models is the very large number of variables to be processed. For each reaction modeled by the extended Arrhenius law, three variables are subject to modification. Thus, a methane combustion mechanism such as the GRI-Mech, containing 325 reactions, potentially involves the modification of a thousand variables. Establishing the optimal values of the unknown parameters represents a very complex inverse problem.

Different numerical approaches have been developed to address this type of problem. Gradient-based optimization has been used to optimize low dimensional kinetic mechanisms containing a few tens of reactions.[7,8] Such methods can be effective in solving small scale systems. Nevertheless, gradient based methods are tailored to the search for the nearest optimum and the objective functions of the problems in chemical kinetics modelling are generally complex, having multiple ridges and valleys, especially if the dimension of the kinetic models is large. Therefore, the success of the method is very dependent on the initial guess since the algorithm can easily get trapped in a local optimum.

On the other hand, evolutionary algorithms are particularly well suited to finding the optimum for chemical reaction systems having a large number of variables with a highly structured response surface.[9] These optimization tool of metaheuristic nature are also called "evolutionary population based method". These algorithms are stochastic and can be used as a local or global search procedure, tackling optimization problems of non-linear and potentially noisy multimodal functions.[10,11]

The Genetic Algorithm (GA) is an optimization method inspired by the theory of evolution. It is regularly used for optimizing kinetic models and represents a very efficient method for optimizing kinetic models.[12–14] In their review, Elliot et al.[15] presented the efficiency of a genetic algorithm for the optimization of hydrogen, methane and kerosene mechanisms, based on experimental datasets. In addition to a reduction process, this algorithm can also be exploited in order to compensate for the errors resulting from the loss of information[16–19].

Particle Swarm Optimization (PSO) is another evolution algorithm. It consists of exploiting the swarm intelligence that relies on the independent evolution of particles and their interactions in a biological type system. Although this algorithm is also well adapted to the optimization of large problems, it has been rarely used for the optimization of kinetic models.[20–23] Ding et al.[22] proved the better optimization performances of PSO compared to GA when applied for three-component parallel reaction mechanism of biomass pyrolysis. However, they used only one set of parameters (further referred as optimization "strategy"), which limits the comparison. Guo et al.[20] compared different PSO strategies to the genetic algorithm (for which only one strategy was used). They showed that, although the PSO approach is prone to fall into a local optimal prematurely, the adaptation of the PSO coefficients could allow better optimization performance than the genetic algorithm. However, and based on the authors' knowledge, this tool has not been used for the optimization of large detailed kinetic mechanisms (i.e. with a number of reactions much higher than ten), so far. The efficiency of this algorithm has been proved in many other different fields[24–31].

The main objective of our current study is to evaluate the efficiency of the Particle

Swarm algorithm for the optimization of detailed kinetic mechanisms. The performances will be assessed and compared to those of the genetic algorithm. The optimization is performed using an open-source tool based on Python (Brookesia[32]) and the well-known methane kinetic model GRI-Mech 3.0.[33]

2 Method

2.1 Computing

2.1.1 Simulation conditions and degraded mechanism settings

The conditions evaluated in the present study are composed of six cases of reactor at atmospheric pressure (constant enthalpy) for CH_4/air mixtures at the equivalence ratio $\Phi = 0.5, 1, 1.5$ of initial temperature 1050 K and 1500 K. Simulations are performed with GRI-Mech 3.0[33] and the Cantera solver[34]. Subsequent evaluation of important reactions and optimizations are performed with Brookesia.[32]

In order to evaluate the ability of optimization algorithms to improve the predictivity of a detailed kinetic model, the GRI-Mech3.0. mechanism has been "degraded". To do this, the kinetic constants of the 49th "most important" reactions were randomly modified. The importance of the reactions is evaluated on the basis of the direct interaction coefficients between a species A and a reaction r_i, that are calculated according to the formulation of Pepiot-Desjardin and Pitch[35] as:

$$r_{Ar_i} = \frac{|\nu_{i,A}\omega_i|}{\max\left(P_A, C_A\right)} \tag{1}$$

with:

$$\begin{aligned} P_A &= \sum_{i=1}^{n_{\text{reactions}}} \max(0, \nu_{i,A}\omega_i) \\ C_A &= \sum_{i=1}^{n_{\text{reactions}}} \max(0, -\nu_{i,A}\omega_i) \end{aligned} \tag{2}$$

$\nu_{i,A}$ represents the stoichiometric coefficient of species A in reaction i and ω_i the net reaction rates of the ith reaction.

The optimization targets are CH_4, CO, CO_2, C_2H_4, C_2H_6 concentration profiles, the temperature and the ignition delay time. To calculate the interaction coefficients for the temperature and ignition delay time targets, CO_2 and CH_3 species were considered, respectively. Seven reactions with the highest interaction coefficients have been selected for each species, bringing the total number of modified reactions to 49.

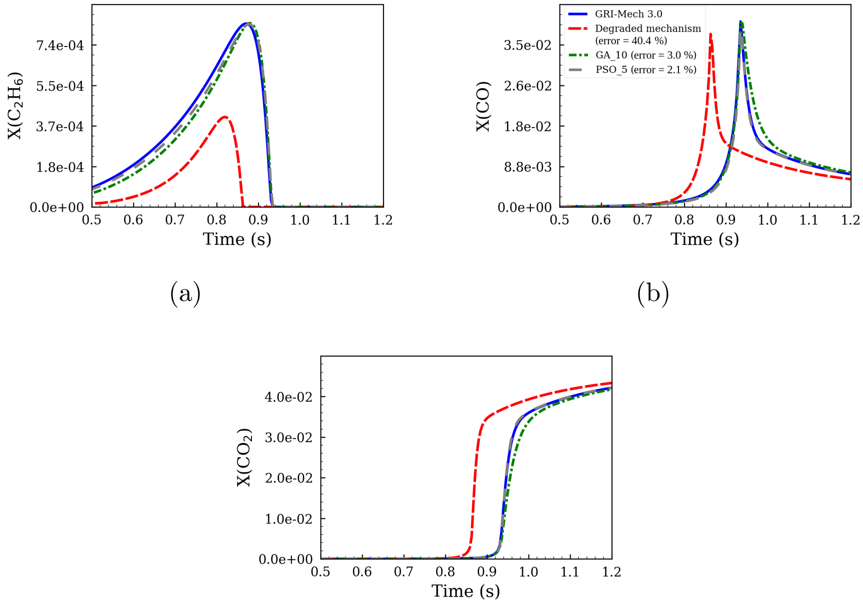
By only modifying the important reactions for the targets, it is possible not to affect reactions having no, or very little, impact on the simulation results and whose constants could be strongly modified without control. The variations applied to the terms B, n, and C of the extended Arrhenius law, expressed in eq. 3 are 15, 5, and 5 %, respectively.

$$k = B \cdot T^n \exp(\frac{-C}{RT}) \tag{3}$$

Figure 1 shows an example of a reactor simulation carried out using the initial mechanism and then using the degraded mechanism. The differences between the results of the two simulations, and thus the loss of predictivity of the degraded mechanism are clearly observed.

2.1.2 Evaluation of the optimized mechanisms quality

For each optimized mechanism and at each iteration, the relative errors of the targets are calculated and averaged. The relative ignition time error is calculated as the difference between the ignition time obtained by the reference and the optimized mechanism divided by the time obtained by the reference. Concerning the variables evolving over time (temperature, species), errors are calculated as the relative differences between the areas under the curves obtained with the reference simulations and the simulations resulting from the optimized mechanisms. Details on the calculations are provided in the documentation of Brookesia.[32]



(a)

(b)

(c)

Figure 1: C_2H_6, CO, and CO_2 molar fractions evolution computed for CH_4/air mixtures $\Phi =$ 0.5 and 1500 K. Simulations performed with the GRI-Mech 3.0 mechanism, the "degraded" version, then "optimized" versions via the GA and PSO algorithms.

Initially, the averaged error calculated for the *degraded* mechanism is 40.4 %.

The fitness value for the optimization algorithms is calculated as the inverse of the averaged error. In doing so, the best individuals have the highest fitness.

2.1.3 Considerations for the comparison of stochastic optimization methods

The global search algorithms addressed in this work are stochastic in nature and therefore the quality of the optimization depends on many random parameters such as the quality of the first generation. In a stochastic approach, the output possesses some inherent randomness. The same set of input parameters values and initial conditions can lead to a set of different outputs. Thus, to best evaluate the differences between the optimization schemes, each strategy was evaluated 20 times and the curves were averaged.

It is important to note that the set of GA and PSO strategies has been developed so that at each iteration, an equal number of simulations is performed. This prerequisite ensures that the algorithms can be compared for an equal numerical cost.

2.2 Genetic algorithm

The genetic algorithm, developed by Holland in the 1960s,[36] is a stochastic optimization approach inspired by Darwin's principle of natural selection. It can be used in the application of inverse methods, where the solution to be reached is known but the input parameters are not, or not accurately. The algorithm will then consist of performing many simulations with different sets of input parameters. At the end of this series of simulations, the sets of parameters giving the results closest to the optimal solution are retained (Selection). Operators will have the function of interchanging these parameters or of crossing them by means of operations such as averages, weighted averages, etc. (Cross-Over), or of randomly modifying one or more variables (Mutation) to extend the search as far as possible in the space of the solutions and to avoid as far as possible the trapping of the optimum search in a local minimum. A more complete description of different operators can be found in the reference.[37]

The GA algorithm coded in the Brookesia software integrates 4 selection operators (Roulette, Rank, Geometric Norm, and Elitism), 4 cross-over operators (Simple, Multiple, Arithmetic, Heuristic), and 4 mutation methods (Uniform, Non-uniform, Boundary). Details on the application of each of these operators are provided in the Brookesia documentation.[32]

To evaluate the potential of the GA algorithm, the impact of each module was evaluated separately. The table 1 summarizes the choice of operators for the 11 evaluated strategies. Strategies 1 to 4 each uses a different selection method. Strategies 5 to 8 / 9 to 11 assess the efficiency of the different crossing/mutation methods.

Table 1: The choice of GA operators for the 11 evaluated strategies

Strategies	**Selection**				**Crossover (*)**				**Mutation (*)**		
	Roulette	Rank	Geometric norm	Elitism	Simple	Multiple	Arithmetic	Heuristic	Uniform	Non-uniform	Boundary
1	**x**				2	2	4	4	3	3	2
2		**x**			2	2	4	4	3	3	2
3			**x**		2	2	4	4	3	3	2
4				**x**	2	2	4	4	3	3	2
5			x		**6**	**2**	**2**	**2**	3	3	2
6			x		**2**	**6**	**2**	**2**	3	3	2
7			x		**2**	**2**	**6**	**2**	3	3	2
8			x		**2**	**2**	**2**	**6**	3	3	2
9			x		2	2	4	4	**6**	**1**	**1**
10			x		2	2	4	4	**1**	**6**	**1**
11			x		2	2	4	4	**1**	**1**	**6**

2.3 PSO

2.3.1 Principle

This algorithm was proposed by Russel Eberhart (electrical engineer) and James Kennedy (socio-psychologist) in 1995.[38] Particle Swarm Optimization is a population-based, self-adaptive research method inspired by the social behavior of groups of birds or fish schools. The PSO algorithm explores the search space, thanks to a population of particles interacting with each other and adapting their position and speed during the optimization process. Research of the optimum is guided by a local research method, based on personal experience, and complemented by a global research method, based on the experience of the group, trying to balance exploration and exploitation.

The dynamic of a particle i is thus governed by an "inertial" component, relative to its speed at the previous iteration t, a "cognitive" component based on the best position P_i found by the individual during the optimization process and a "social" component based on the best location of the population found at the iteration t, P_g. In a hyperspace of dimension d, the velocity and position of a particle i are, respectively, governed by the following equations:

$$v_{i,d}(t+1) = \underbrace{W \cdot v_{i,d}(t)}_{\text{inertial component}} + \underbrace{C_1 \cdot a_1 \cdot (P_{i,d}(t) - x_{i,d}(t))}_{\text{cognitive component}} + \underbrace{C_2 \cdot a_2 \cdot (P_{g,d}(t) - x_{i,d}(t))}_{\text{social component}} \tag{4}$$

$$x_{i,d}(t+1) = x_{i,d}(t) + v_{i,d}(t+1) \tag{5}$$

where W represents the inertia weight, C_1, C_2 represent the acceleration constants, and a_1, a_2 are values, ranging from 0 to 1, randomly set at each PSO iteration t and for each particle i.

2.3.2 Constriction factor

The optimization of parameters by PSO algorithm was initially of an empirical nature. In 1999, the work of Clerc[39] demonstrated the necessity to use a constriction factor to ensure the convergence of the particle swarm algorithm. A new law, incorporating the constriction term, governs the velocity of a particle by the following equation:[40]

$$v_{i,d}(t+1) = \chi \cdot [v_{i,d}(t) + c_1 \cdot a_1 \cdot (P_{i,d}(t) - x_{i,d}(t)) + c_2 \cdot a_2 \cdot (P_{g,d}(t) - x_{i,d}(t))] \tag{6}$$

where χ is the constriction factor, such as:

$$\chi = \frac{2\kappa}{\left|2 - \phi - \sqrt{\phi^2 - 4\phi}\right|} \tag{7}$$

with $\phi = c_1 + c_2$, $\phi \geqslant 4$, $\kappa \in [0, 1]$.

In this equation, the variable κ controls the speed of convergence: fast convergence to a stable point shall be obtained when κ equals 0, while a κ equal to 1 results in slower convergence.

2.3.3 Evolution of different components weights during the iterative optimization process

The success of a population-based algorithm depends on the subtle balance between local and global searching. This equilibrium must be managed globally for the entire optimization and can also be refined as the process progresses. Thus, it is possible to vary the values of the parameters w, c_1, and c_2 to favor a global search at the beginning of the optimization and then the local search at the end. These approaches are discussed in section 2.3.6.

2.3.4 Score calculation

Other approaches have been developed to improve the efficiency of the algorithm in more specific contexts. For example, Cai et al.[41] introduced a score variable S to rank particles according to their fitness, following this equation:

$$S_i(t) = \frac{f_i(t) - f_{\text{worst}}(t)}{f_{\text{best}}(t) - f_{\text{worst}}(t)} \tag{8}$$

This variable can be used to weight the inertial component: by affecting a high inertia to a particle having a low fitness level, the global research is promoted. Conversely, assigning a low inertia to a particle with a high fitness level favors the local search for the particles closest to an optimum (which is a maximum in the current study). The weighted inertia of the particle i ranges between a maximum limit value $W_{\max}$ and a minimum limit value $W_{\min}$:

$$W_i(t) = W_{\min}(t) + (W_{\max}(t) - W_{\min}(t)) \cdot (1 - S_i(t)) \tag{9}$$

2.3.5 Convergence detector

The PSO algorithm implemented in Brookesia integrates a convergence detector. If the standard deviation between the fitness levels of the whole population is less than 10 % of the maximum fitness value, the direction of the inertial component is momentarily multiplied by a factor of -5 or +5:

$$W_i(t) = W_i(t) \cdot \begin{cases} 1 \;\text{ if }\; \sigma(\mathbf{f}) \geqslant 0.1 \cdot f_{\max} \\ 5 \cdot (-1)^{a_3} \quad \text{otherwise} \end{cases} \tag{10}$$

with a_3, a binary random variable on $\{0;1\}$. In this way, the diversity of the population is guaranteed during the optimization process. When 90% of the iterations have been performed, this detector is deactivated in order to enable a local search, more relevant at the end of the optimization process.

Many other tools can be integrated into the PSO code for the improvement of the identification quality, depending on the application.[20,23,29,31,42]

2.3.6 Panorama of PSO strategies

Like any evolutionary algorithm approach, the PSO method is prone to the problems of low convergence, convergence to a local minimum, and so forth. To design the best PSO configuration for detailed mechanism optimization, 14 strategies have been evaluated and compared.

In order to visualize more precisely the optimization schemes, criteria for the relative importance of the inertial, cognitive and social components were defined:

$$\widetilde{W} = \frac{W}{W + (1/2) \cdot (C_1 + C_2)} \tag{11}$$

$$\widetilde{C_1} = \frac{(1/2) \cdot C_1}{W + (1/2) \cdot (C_1 + C_2)} \tag{12}$$

$$\widetilde{C_2} = \frac{(1/2)\cdot C_2}{W + (1/2)\cdot(C_1 + C_2)} \tag{13}$$

Note that the absolute values of the terms C_1 and C_2 have been divided by two to take into account the influence of the random factors a_1 and a_2.

The evolution of these criteria during the optimization process is presented, for each of the strategies described hereafter, in figure 2.

- **Strategies with constriction factor (1 to 4)**

  The first strategy is a reference. It is based on the work of Clerc[39] who demonstrated the necessity to use a constriction factor to ensure the convergence of the particle swarm algorithm (see equation 6). The parameters of the equation are often set as $c_1 = c_2 = 2.05$ and $\kappa = 1$.[40,43,44] In such a case, the optimization scheme is similar to the equations 4 and 5 with the parameters $W = 0.730$ and $C_1 = C_2 = 1.496$.

  The second strategy proposes to keep the same values for the terms c_1 and c_2. On the other hand, the value of κ is divided by 2 ($\kappa = 0.5$) to evaluate the effect of an *a priori* faster convergence.

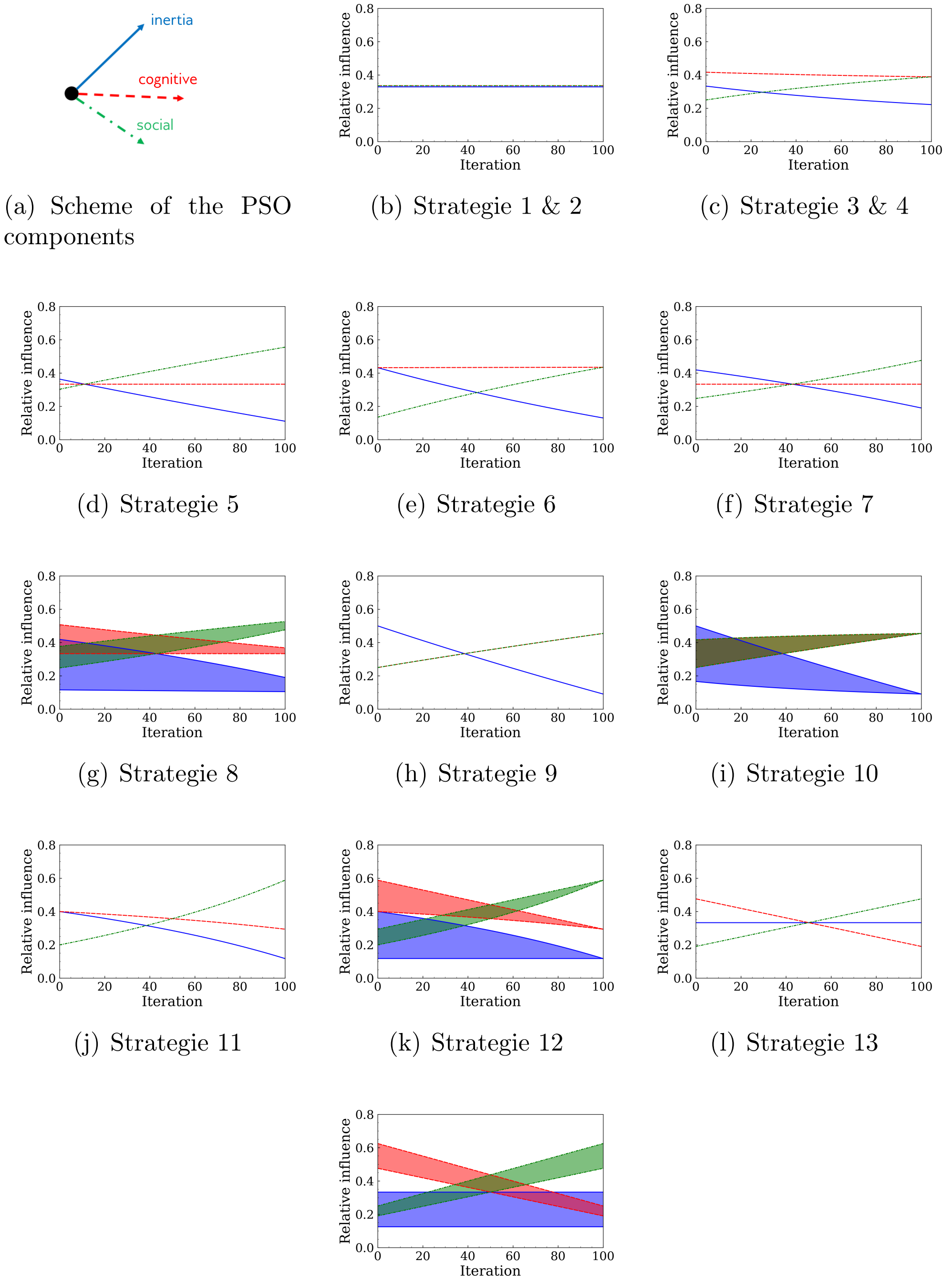
  c_1 and c_2 parameters evolve from 2.5 to 3.5 and 1.5 to 3.5, respectively, in strategies 3 and 4 to favor a local search at the end of the optimization process. The value of κ is 1 in strategy 3 and is reduced to 0.5 in strategy 4.

- **Strategies without constriction factor, with decreasing inertia, constant cognitive component, and increasing social component (5 to 8)**

  To favor a global search at the beginning of the optimization then the local search at the end, strategies 5 to 8 and the corresponding parameters w, c_1 and c_2 have been defined such that the relative influence of the inertial component (favorable to the global search) decreases over iterations, the cognitive component remains constant, and the social component (favorable to the local search) increases. On this basis, several patterns are possible according to the initial importance given to each of the components at the beginning of the process. It is to be noted that strategy 8 is composed of the same parameters as strategy 7 but the inertia term W is weighted by the individual's score (see equation 8).

- **Strategies without constriction factor, with decreasing inertia, decreasing cognitive component, and increasing social component (9 to 12)**
  In these strategies, the relative weights of inertia and cognitive components decrease over the iterations while that of social component increases which favors local search as long as the optimization progresses. The parameters of strategies 10 and 12 are similar to those of strategies 9 and 11, respectively, but the inertia term W is weighted by the individual's score.

- **Strategies without constriction factor, with constant inertia, decreasing cognitive component, and increasing social component (13 to 14)**
  In these strategies, the relative importance of inertia remains constant, which maintains a non-negligible share of randomness throughout the optimization. To tend towards a local search, the weight of the cognitive and social components decreases and increases, respectively. Strategy 14 integrates the individual's score in the calculation of the inertial weight.



(a) Scheme of the PSO components

(b) Strategie 1 & 2

(c) Strategie 3 & 4

(d) Strategie 5

(e) Strategie 6

(f) Strategie 7

(g) Strategie 8

(h) Strategie 9

(i) Strategie 10

(j) Strategie 11

(k) Strategie 12

(l) Strategie 13

(m) Strategie 14

Figure 2: Relative importance of the different components in the strategies adopted for optimization by the PSO method

Table 2: Selected parameters for the development of PSO strategies

<table>
<tr><th rowspan="2">Strat</th><th colspan="2" rowspan="2">κ</th><th colspan="2">c1</th><th colspan="2">c2</th><th></th></tr>
<tr><th>init</th><th>final</th><th>init</th><th>final</th><th></th></tr>
<tr><td>1</td><td colspan="2">1</td><td>2.05</td><td>2.05</td><td>2.05</td><td>2.05</td><td></td></tr>
<tr><td>2</td><td colspan="2">0.5</td><td>2.05</td><td>2.05</td><td>2.05</td><td>2.05</td><td></td></tr>
<tr><td>3</td><td colspan="2">1</td><td>2.5</td><td>3.5</td><td>1.5</td><td>3.5</td><td></td></tr>
<tr><td>4</td><td colspan="2">0.5</td><td>2.5</td><td>3.5</td><td>1.5</td><td>3.5</td><td></td></tr>
<tr><th rowspan="2">Strat</th><th colspan="2">W</th><th colspan="2">C1</th><th colspan="2">C2</th><th>Score</th></tr>
<tr><th>init</th><th>final</th><th>init</th><th>final</th><th>init</th><th>final</th><th>Wmin</th></tr>
<tr><td>5</td><td>0.6</td><td>0.2</td><td>1.1</td><td>1.2</td><td>1</td><td>2</td><td></td></tr>
<tr><td>6</td><td>0.8</td><td>0.3</td><td>1.6</td><td>2</td><td>0.5</td><td>2</td><td></td></tr>
<tr><td>7</td><td>1.1</td><td>0.4</td><td>1.75</td><td>1.4</td><td>1.3</td><td>2</td><td></td></tr>
<tr><td>8</td><td>1.1</td><td>0.4</td><td>1.75</td><td>1.4</td><td>1.3</td><td>2</td><td>0.4</td></tr>
<tr><td>9</td><td>1</td><td>0.2</td><td>1</td><td>2</td><td>1</td><td>2</td><td></td></tr>
<tr><td>10</td><td>1</td><td>0.2</td><td>1</td><td>2</td><td>1</td><td>2</td><td>0.2</td></tr>
<tr><td>11</td><td>1</td><td>0.2</td><td>2</td><td>1</td><td>1</td><td>2</td><td></td></tr>
<tr><td>12</td><td>1</td><td>0.2</td><td>2</td><td>1</td><td>1</td><td>2</td><td>0.2</td></tr>
<tr><td>13</td><td>0.7</td><td>0.7</td><td>2</td><td>0.8</td><td>0.8</td><td>2</td><td></td></tr>
<tr><td>14</td><td>0.7</td><td>0.7</td><td>2</td><td>0.8</td><td>0.8</td><td>2</td><td>0.2</td></tr>
</table>

3 Results & Discussion

3.1 Convergence curves comparison

3.1.1 Genetic algorithm

The convergence curves obtained using the genetic algorithm strategies are presented in graph a) of figure 3. The evolution of the mean error found for the estimation of the optimization targets is presented as a function of the number of iterations. It is important to remember here that each curve presented in the figure is the average result of 20 independent optimizations (see section 2.1.3). This approach intends to take into account the stochastic nature of the optimization algorithms and to ensure rigorous comparisons.

The assessment of the different strategies for the genetic algorithm shows an important sensitivity to the selection method. The algorithm efficiency with rank-based selection is much worse than that obtained with other methods. The selection by geometric norm presents slightly inferior results compared to the selection by roulette and elitism. Nevertheless, considering the standard deviations between the optimizations (discussed in section 3.1.3), the performances can be considered as quite similar.

The GA_5 to GA_8 strategies differ in the importance devoted to the different crossover operators. The convergence curves presented in figure 3 a) show that the quality of optimization is very slightly affected by these choices. The choice of the mutation method, on the other hand, has a greater impact. The results indicate that the quality of optimization is higher when non-uniform or boundary mutation operators are predominantly used.

The GA_9 to GA_11 strategies have different importances in the use of mutation operators. Mutation consists of randomly varying the variables to be optimized. In the Brookesia code, about 30 % of the variables are modified. Note that the non-uniform mutation takes into account the progress of the optimization process. Thus, the more the number of iterations evolves, the less the modifications on the variables are important (more details on these methods are provided in the documentation of Brookesia[32]). This allows to maximize the exploration of the search space at the beginning of the optimization and to focus on a local search at the end. The comparison shows that, in the context of the optimization of detailed kinetic models, the latter method should be preferred to uniform mutation. It should also be noted that an important use of the boundary mutation operator also gives good results. Nevertheless, the use of this method leads to a large increase in the standard deviation between optimizations (see figure 4) indicating that the quality of the optimization can vary significantly from one run to another.

This work highlighted the sensitivities in the choice of genetic algorithm operators for the optimization of detailed kinetic mechanisms. The main observations are that the algorithm is not very sensitive to the choice of selection operators (except the rank selection) and crossing operators and is more sensitive to the choice of mutation, for which the non-uniform operator should be preferred. Moreover, the convergence curves established for a large number of GA strategies provide a consistent basis for comparison to evaluate the efficiency of the PSO method, which has not, to the authors' knowledge, been used for the optimization of detailed kinetic mechanisms, so far.

3.1.2 Particle Swarm Optimization

The convergence curves for the PSO strategies are presented in figure 3 b). Strategies 1 to 4 integrate the implementation of a constriction factor to ensure theoretical convergence. Strategies 1 to 4 integrate the implementation of a constriction factor to ensure theoretical convergence. As illustrated in figure 2, the relative influences of the inertial, cognitive and social components are constant for strategies 1 and 2 and are evolutive for strategies 3 and 4. Strategies 2 and 4 are affected by a constriction factor equal to 0.5, instead of 1 used for strategies 1 and 3, to accelerate the speed of convergence. It is noticeable that the distinction of the optimization quality between these four strategies is made almost exclusively on the choice of the convergence speed. Thus, strategies 2 and 4 present significantly better results than strategies 1 and 3. Given the very large number of variables to be optimized in a detailed mechanism optimization problem, it appears more appropriate to favor rapid convergence.

The set of strategies 5 to 8 gives, surprisingly, very different results and integrates both the strategy with the highest performance (6) and the least effective strategy (8). A more global analysis of the strategies seems to show that, for this type of problem, the higher relative importance of the cognitive component has a positive influence on the optimization, whereas the high importance of the inertial component is in contrast quite unfavorable. Thus, strategies 7 and 8, which have a strong inertial weight at the beginning of the optimization, give poorer results. We also note, when comparing these two strategies, that the score parameter integrated into the strategy 8, and which was intended to promote global research for individuals with low fitness levels, has an unfavorable effect on the quality of the optimization. Interestingly, this difference, although relatively small, is found for all pairs of strategies with/without score (7/8, 9/10, 11/12, and 13/14).

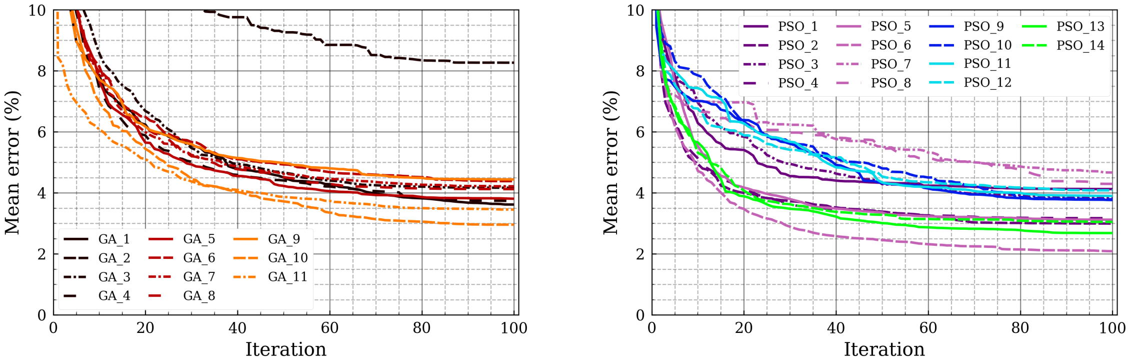
The comparison between the pairs of strategies 9/10 and 11/12 shows that the inversion of the weight evolution for the cognitive component during the optimization process while keeping W and C2 coefficients equal, has little effect on the quality of the optimization. Besides, the pair of strategies 13/14, with coefficients C1 and C2 comparable to those of pair 11/12 and whose influence of the inertial component is modified (less important initially but higher at the end) provides more efficient optimizations.

3.1.3 Comparison between GA and PSO strategies

A global comparison of the set of strategies shows that, overall, the PSO algorithm provides better optimization than the GA. Thus, among the 5 best strategies, 4 are obtained by the PSO algorithm while among the 5 least efficient, 3 were obtained when the GA algorithm was applied. Nevertheless, the efficiency of optimization is more variable between PSO strategies. Thus, except for strategy 2, which stands out from the other trends, the average error at the end of optimization by GA is between 3 and 4.5 %. It is between 2 and 5 % for the PSO algorithm. This indicates a higher sensitivity to the choice of optimization parameters when applying the PSO algorithm.

The convergence rate of the PSO algorithm seems significantly faster than that of the genetic algorithm. Thus, the residual error for the 6 best PSO strategies drops to about 4 % or less after the first 20 generations, whereas equivalent performance is obtained after 40 generations for GA strategies 10 and 11.

Figure 4 shows the standard deviations calculated between the 20 optimizations performed for each strategy. Overall, the standard deviations of the PSO algorithm are higher than those of the GA. Excluding the value of GA_2, the mean standard deviation is 16 % lower for GA than for PSO (1.303 versus 1.550, respectively). Thus, among the 5 highest standard deviation values, 4 were measured with PSO methods (strategies 1, 7, 8, and 9) and only one with the genetic algorithm (strategy 7). These results indicate a higher variability of optimization qualities for the same strategy. It is worth noting that larger values of standard deviations mean lower reliability of the strategy.



(a) Genetic Algorithm

(b) PSO algorithm

Figure 3: Mean error evolution as a function of the iteration number for the different optimization strategies tested

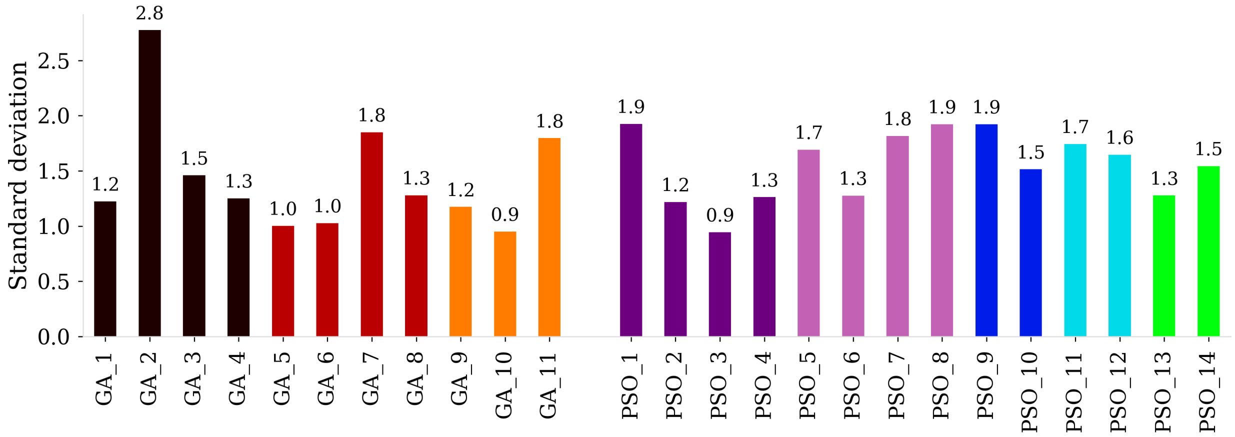


Figure 4: Standard deviation between optimization processes

3.2 Comparison of mechanisms

As a reminder, 49 reactions of the GRI-Mech 3.0 mechanism were randomly modified to build a *degraded* version. The variation of the coefficients B, n, and C ranged up to 15, 5, and 5 %, respectively. The degraded version was then optimized to recover the predictive character of the GRI-Mech mechanism.

Figure 5 shows the mean relative deviation of the B, n, and C coefficients of the modified reactions between the GRI-Mech 3.0 mechanism, the degraded version, and then the optimized versions. For all the strategies tested in this work, the optimization reduces the differences between the kinetic constants of the degraded version and the reference mechanism.

This observation underlines the reliability of the optimization approach using evolutionary algorithms. It can also be noted that the difference magnitude between the kinetic constants of the optimized mechanism and the GRI-Mech 3.0 mechanism reflects, overall, the optimization quality. Thus, for the most efficient optimizations identified in the previous section, the difference amplitudes are among the smallest and can be reduced by up to a factor of 3 (PSO strategies 5, 6, 13, and 14, for example) compared to the degraded mechanism.

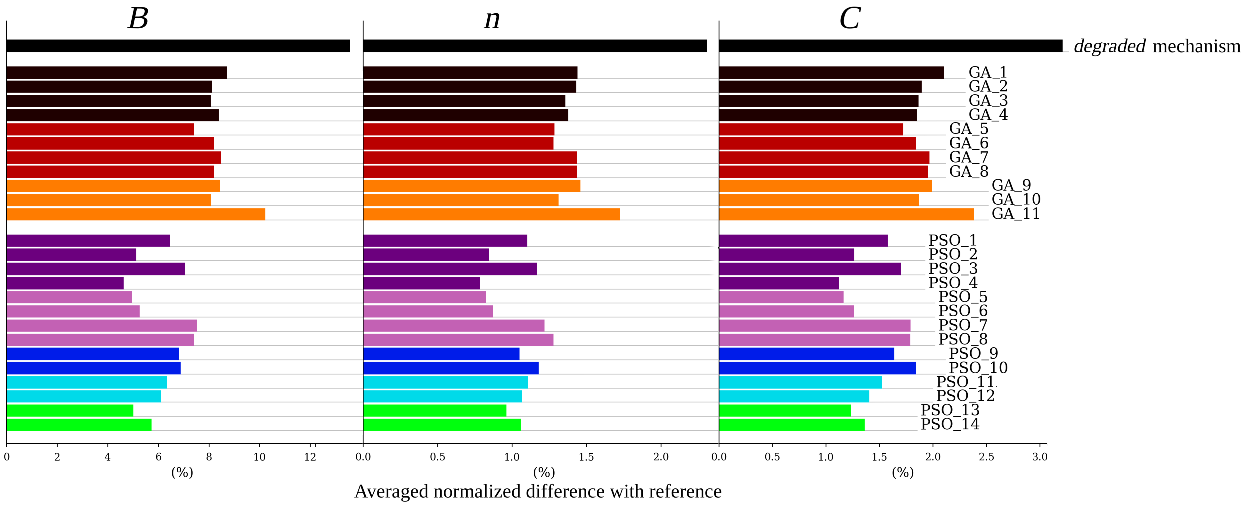


Figure 5: Standard deviation between optimization processes

As indicated in paragraph 2.1.3, the initial error measured between the profiles simulated by the degraded mechanism and the GRI-Mech 3.0 reference mechanism is 40.4 %. For the set of strategies presented here, this error is reduced up to about 4 % after the optimization process. It is reduced on average to 3.0 % after optimization by GA with strategy 10, and to 2.1 % by strategy 6 of the PSO approach. As an illustration, the simulations obtained by optimized mechanisms using the two strategies are presented in figure 1. The comparison clearly illustrates the improved quality of the kinetic model after optimization in both cases, since a very good agreement is obtained between the simulation made with the optimized models and the reference mechanism.

4 Conclusion

The purpose of this work was to evaluate the potential of the Particle Swarm Optimization (PSO) algorithm for the optimization of detailed kinetic mechanisms. To this end, different strategies were implemented to optimize an altered version of the GRI-Mech mechanism. PSO was compared to the genetic algorithm (GA), which is nowadays commonly used for the optimization of kinetic models. The results show that the best strategies of the algorithm offer higher performance than those obtained for the genetic algorithm. The speed of convergence observed for a large number of PSO approaches is also significantly faster. Besides, the evolution of the optimized kinetic constants converges, for all the strategies tested in this work, towards the initial values of the GRI-Mech 3.0 mechanism, which underlines the relevance of the approaches used. The statistical analysis of the results showed that the quality of the optimizations can vary significantly from one optimization to another. It appears that the dispersion of the results is generally higher with PSO than with GA.

In conclusion, this work attests to the good performance of the PSO algorithm compared to the equivalent GA method, which is much more widely used for the optimization of kinetic mechanisms. Given the simplicity of the algorithm, this work demonstrates the interest in considering Particle Swarm Optimization as a good alternative to the Genetic Algorithm, for such applications.

References

(1) Lu, T.; Law, C. K. Toward accommodating realistic fuel chemistry in large-scale computations. *Prog. Energy Combust. Sci.* **2009**, *35*, 192–215.

(2) Javoy, S.; Mével, R.; Dupré, G. Oxygen atom kinetics in silane-hydrogen-nitrous oxide mixtures behind reflected shock waves. *Chem. Phys. Lett.* **2010**, *500*, 223–228.

(3) Carpenter, B. K. Ab Initio Computation of Combustion Kinetics. 1. Vinyl Radical + O2. *The J. Phys. Chem.* **1995**, *99*, 9801–9810.

(4) Sarathy, S. M.; Oßwald, P.; Hansen, N.; Kohse-Höinghaus, K. Alcohol combustion chemistry. *Prog. Energy Combust. Sci.* **2014**, *44*, 40–102.

(5) Wang, H.; Sheen, D. A. Combustion kinetic model uncertainty quantification, propagation and minimization. *Prog. Energy Combust. Sci.* **2015**, *47*, 1–31.

(6) Frenklach, M.; Wang, H.; Rabinowitz, M. J. Optimization and analysis of large chemical kinetic mechanisms using the solution mapping method-combustion of methane. *Prog. Energy Combust. Sci.* **1992**, *18*, 47–73.

(7) Bock, H. G. In *Modelling of Chemical Reaction Systems*; Goldanskii, V. I., Schäfer, F. P., Gomer, R., Toennies, J. P., Ebert, K. H., Deuflhard, P., Jäger, W., Eds.; Springer Berlin Heidelberg: Berlin, Heidelberg, 1981; Vol. 18; pp 102–125.

(8) Rabitz, H. Chemical sensitivity analysis theory with applications to molecular dynamics and kinetics. *Comput. & Chem.* **1981**, *5*, 167–180.

(9) Gallagher, K.; Sambridge, M. Genetic algorithms: A powerful tool for large-scale nonlinear optimization problems. *Comput. & Geosci.* **1994**, *20*, 1229–1236.

(10) Michalewicz, Z. *Genetic Algorithms + Data Structures = Evolution Programs*; Springer Berlin Heidelberg: Berlin, Heidelberg, 1996.

(11) Bäck, T., Fogel, D. B., Michalewics, Z., Eds. *Handbook of evolutionary computation*; Institute of Physics Pub. ; Oxford University Press: Bristol ; Philadelphia : New York, 1997.

(12) Balland, L.; Estel, L.; Cosmao, J. M.; Mouhab, N. A genetic algorithm with decimal coding for the estimation of kinetic and energetic parameters. *Chemom. Intell. Lab. Syst.* **2000**, *50*, 121–135.

(13) Rein, G.; Lautenberger, C.; Fernandez-Pello, A. C.; Torero, J. L.; Urban, D. L. Application of genetic algorithms and thermogravimetry to determine the kinetics of polyurethane foam in smoldering combustion. *Combust. Flame* **2006**, *146*, 95–108.

(14) Kadiva, A.; Sadeghi, M. T.; Sotudeh-Gharebagh, R.; Mahmudi, M. Estimation of Kinetic Parameters for Hydrogenation Reactions Using a Genetic Algorithm. *Chem. Eng. & Technol.* **2009**, *32*, 1588–1594.

(15) Elliott, L.; Ingham, D.; Kyne, A.; Mera, N.; Pourkashanian, M.; Wilson, C. Genetic algorithms for optimisation of chemical kinetics reaction mechanisms. *Prog. Energy Combust. Sci.* **2004**, *30*, 297–328.

(16) Elliott, L.; Ingham, D. B.; Kyne, A. G.; Mera, N. S.; Pourkashanian, M.; Whittaker, S. Reaction mechanism reduction and optimisation for modelling aviation fuel oxidation using standard and hybrid genetic algorithms. *Comput. & Chem. Eng.* **2006**, *30*, 889–900.

(17) Perini, F.; Brakora, J. L.; Reitz, R. D.; Cantore, G. Development of reduced and optimized reaction mechanisms based on genetic algorithms and element flux analysis. *Combust. Flame* **2012**, *159*, 103–119.

(18) Jaouen, N.; Vervisch, L.; Domingo, P.; Ribert, G. Automatic reduction and optimisation of chemistry for turbulent combustion modelling: Impact of the canonical problem. *Combust. Flame* **2017**, *175*, 60–79.

(19) Cailler, M.; Darabiha, N.; Veynante, D.; Fiorina, B. Building-up virtual optimized mechanism for flame modeling. *Proc. Combust. Inst.* **2017**, *36*, 1251–1258.

(20) Guo, Z.-C.; Chen, L.-P.; Chen, W.-H. Estimation of kinetic parameters from adiabatic calorimetric data by a hybrid Particle Swarm Optimization method. *Chem. Eng. Res. Des.* **2017**, *122*, 273–279.

(21) Ding, Y.; Zhang, Y.; Zhang, J.; Zhou, R.; Ren, Z.; Guo, H. Kinetic parameters estimation of pinus sylvestris pyrolysis by Kissinger-Kai method coupled with Particle Swarm Optimization and global sensitivity analysis. *Bioresour. Technol.* **2019**, *293*, 122079.

(22) Ding, Y.; Zhang, W.; Yu, L.; Lu, K. The accuracy and efficiency of GA and PSO optimization schemes on estimating reaction kinetic parameters of biomass pyrolysis. *Energy* **2019**, *176*, 582–588.

(23) Sun, S.; Yan, H.; Meng, F. Optimization of a Fluid Catalytic Cracking Kinetic Model by Improved Particle Swarm Optimization. *Chem. Eng. & Technol.* **2020**, *43*, 289–297.

(24) Eberhart,; Yuhui Shi, Particle swarm optimization: developments, applications and resources. Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546). 2001; pp 81–86 vol. 1.

(25) Zwe-Lee Gaing, A particle swarm optimization approach for optimum design of PID controller in AVR system. *IEEE N.a. Energy Convers.* **2004**, *19*, 384–391.

(26) Jin, N.; Rahmat-Samii, Y. Particle Swarm Optimization for Antenna Designs in Engineering Electromagnetics. *J. Artif. Evol. Appl.* **2008**, *2008*, 1–10.

(27) Valle, Y. d.; Venayagamoorthy, G. K.; Mohagheghi, S.; Hernandez, J.; Harley, R. G. Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems. *IEEE N.a. Evol. Comput.* **2008**, *12*, 171–195.

(28) AlRashidi, M. R.; El-Hawary, M. E. A Survey of Particle Swarm Optimization Applications in Electric Power Systems. *IEEE N.a. Evol. Comput.* **2009**, *13*, 913–918.

(29) Mohammadi-Ivatloo, B.; Moradi-Dalvand, M.; Rabiee, A. Combined heat and power economic dispatch problem solution using particle swarm optimization with time varying acceleration coefficients. *Electr. Power Syst. Res.* **2013**, *95*, 9–18.

(30) Khare, A.; Rangnekar, S. A review of particle swarm optimization and its applications in Solar Photovoltaic system. *Appl. Soft Comput.* **2013**, *13*, 2997–3006.

(31) El Rassy, E.; Billaud, Y.; Saury, D. Simultaneous and direct identification of thermophysical properties for orthotropic materials. *Meas.* **2019**, *135*, 199–212.

(32) Matynia, A.; Delaroque, A.; Chakravarty, H. Brookesia: An open-source tool for reduction and optimization of kinetic mechanisms. `https://github.com/Brookesia-py/Brookesia`, 2020; Version 1.6.0.

(33) Smith, G. P.; Golden, D. M.; Frenklach, M.; Moriarty, N. W.; Eiteneer, B.; Goldenberg, M.; Bowman, C. T.; Hanson, R. K.; Song, S.; Gardiner, W. C.; Lissianski, V. V. GRI-Mech 3.0. 1999; `http://combustion.berkeley.edu/gri-mech/version30/text30.html`.

(34) Goodwin, D. G.; Speth, R. L.; Moffat, H. K.; Weber, B. W. Cantera: An Object-oriented Software Toolkit for Chemical Kinetics, Thermodynamics, and Transport Processes. 2018; `https://cantera.org/`.

(35) Pepiot-Desjardins, P.; Pitsch, H. An efficient error-propagation-based reduction method for large chemical kinetic mechanisms. *Combust. Flame* **2008**, *154*, 67–81.

(36) Holland, J. H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*; MIT Press: Cambridge, MA, USA, 1992.

(37) Goldberg, D. *Genetic Algorithms in search optimization and machine learning*, 1st ed.; Addison-Wesley Longman Publishing Co., Inc.: USA, 1989.

(38) Eberhart, R.; Kennedy, J. A new optimizer using particle swarm theory. MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science. Nagoya, Japan, 1995; pp 39–43.

(39) Clerc, M. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406). Washington, DC, USA, 1999; pp 1951–1957.

(40) Eberhart, R.; Shi, Y. Comparing inertia weights and constriction factors in particle swarm optimization. Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512). La Jolla, CA, USA, 2000; pp 84–88.

(41) Cai, X.; Cui, Z.; Zeng, J.; Tan, Y. In *Particle Swarm Optimization*; Lazinica, A., Ed.; InTech, 2009; pp 89–112.

(42) Lee, K. H.; Baek, S. W.; Kim, K. W. Inverse radiation analysis using repulsive particle swarm optimization algorithm. *Int. J. Heat Mass Transf.* **2008**, *51*, 2772–2783.

(43) Hatanaka, T.; Korenaga, T.; Kondo, N.; Uosaki, K. *Particle Swarm Optim.*; 2009.

(44) Tuppadung, Y.; Kurutach, W. Comparing nonlinear inertia weights and constriction factors in particle swarm optimization. *Int. J. Knowledge-based Intell. Eng. Syst.* **2011**, *15*, 65–70.

Author Contributions

- Code development: Alexis Matynia, Aurélie Delaroque, and Harish Chakravarty
- Algorithm evaluation and testing: Elissa El-Rassy, Patrick Sambou, and Alexis Matynia
- Writing and proofreading of the paper: all co-authors

Notes

Conflicts of Interest: None.

Acknowledgements

The authors gratefully acknowledge financial support from ED SMAER doctoral school of Sorbonne University as well as the ONERA grant No. F/12893DA-GGUE.

TOC

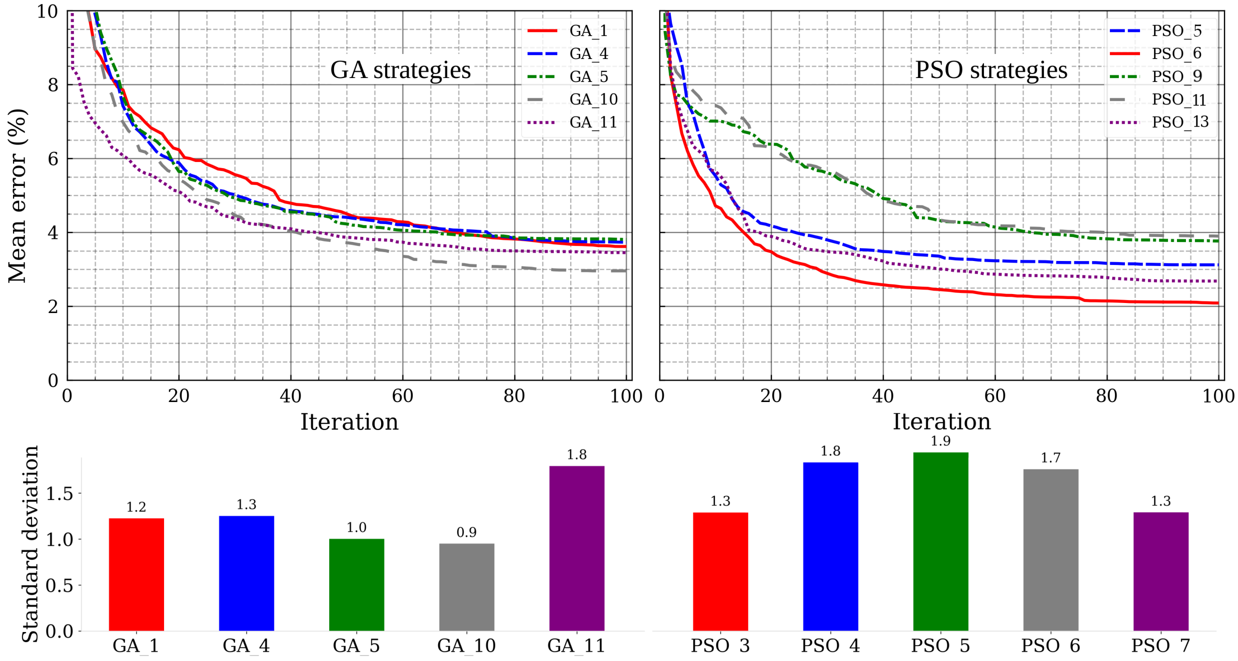


Figure 6: For Table of Contents Only