Applied Soft Computing 143 (2023) 110401

Contents lists available at ScienceDirect

Applied Soft Computing

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

# A particle swarm optimization algorithm with novelty search for combustion systems with ultra-low emissions and minimum fuel consumption

David Martínez-Rodríguez [a], Ricardo Novella [b], Gabriela Bracho [b,*], Josep Gomez-Soriano [b], Cassio Fernandes [b], Tommaso Lucchini [c], Augusto Della Torre [c], Rafael-J. Villanueva [a], J. Ignacio Hidalgo [d,e]

[a] Instituto Universitario de Matemática Multidisciplinar, Universitat Politècnica de València, Spain
[b] Centro de Motores Térmicos, Universitat Politècnica de València, Spain
[c] Politecnico di Milano, Italy
[d] Department of Computer Arquitecture, Universidad Complutense de Madrid, Spain
[e] Instituto de Tecnología del Conocimiento, Universidad Complutense de Madrid, Spain

ARTICLE INFO

*Article history:*
Received 26 September 2022
Received in revised form 16 January 2023
Accepted 3 May 2023
Available online 12 May 2023

*Keywords:*
PSO
NS
Optimization algorithm
CI engine
Combustion
Emissions
Efficiency

ABSTRACT

The particle swarm optimization algorithm is primarily inspired by the natural behaviour of swarms and achieves important results in different applications. However, it is not exempt from stagnation in local optima and has a tendency to prematurely converge to them. Novelty Search is a concept that appeared recently in different fields of computational intelligence. It aims at exploring non-visited areas of the search space through solutions that bring novelty to already discovered solutions. The novelty of this work can be divided into two steps: on one side, this article proposes a variant of the particle swarm optimization algorithm which uses Novelty Search concepts to improve the algorithm's performance. Our proposal is first checked and compared using the CEC 2005 benchmark suite and then, we apply it to solve a real-world optimization problem: the design of a combustion system targeting the reduction of pollutant emissions and fuel consumption. The combustion chamber design phase usually is a complex and time-consuming process even with advanced supercomputers, since it depends on several input variables which are highly non-linear and with crossed interaction. Then, the second contribution of this work is to develop a methodology that couples a computational fluid dynamics (CFD) simulation tool with the new optimization algorithm for minimizing the specific fuel consumption of a compression-ignited engine, while constraining the $NO_x$ and soot emissions. A 3D-CFD model of the combustion system was built to predict and analyse the performance of the combustion system and hence, select the parameters with a higher impact on the system. The method reduces the computational time and includes tools for the automatic preparation of the input parameters and geometry of the system. The input parameters correspond to geometrical variables that control the bowl shape, the number of holes in the injector, the injection pressure, the swirl number and the exhaust gas recirculation rate. Results show how the simulation tool and the new PSO with Novelty Search algorithm allow us to obtain a new combustion system that minimizes the fuel consumption by 3%, simultaneously reducing $NO_x$ and soot emissions.



## 1. Introduction

Particle Swarm Optimization (PSO) [1] has become a powerful search and optimization tool. In the literature we can find many applications for both real world problems [2–4] and theoretical studies [5–7]. PSO does not work well for all types of problems and sometimes presents the usual disadvantages of other bio-inspired algorithms, such as premature convergence, dependence on configuration parameters or obtaining local optimum solutions. For this reason, a whole set of variants have been developed to solve the aforementioned problems.

Following the classification of Gou et al. [8], recent studies mainly focused on improving the performance of PSO can be classified into three groups:

* Correspondence to: CMT Motores Térmicos. Universitat Politècnica de Valencia Valencia, E-46022, Spain.
*E-mail address:* gbracho@mot.upv.es (G. Bracho).

https://doi.org/10.1016/j.asoc.2023.110401
1568-4946/

1. the study and selection of algorithm parameters (number of particles, evaluations, rates and probabilities, etc.),
2. the modification of the structure and topology of communication between particles, and,
3. the development of hybrid algorithms that combine PSO with other heuristics, seeking to improve the exploration of solution space and the exploitation of promising solutions.

In this paper we propose an improvement of PSO, by working on both, the second and the third of the above groups. Specifically, we are investigating the performance of a new hybrid implementation of PSO, in which we propose the use of Novelty Search concepts [9] to improve exploration and exploitation. We call this variant Novelty Swarm (NS) optimization algorithm. Although the definition of memetic computing suffered some modifications since the first appearance of the term [10], three main types of memetic computing algorithms are defined by the Memetic Computing Journal. Following this taxonomy we classify our proposal as a type 1 memetic computing algorithm, where a general-purpose algorithm such as PSO is integrated with the novelty search heuristic.

Novelty Search is a recent paradigm in evolutionary and bio-inspired optimization algorithms developed by J. Lehman and K. O. Stanley. It is based on the idea that fitness function-based algorithms "may actively misdirect search towards dead ends" [11], this is, getting stuck in local optima due to the gradient of the objective function. In population based bio-inspired algorithms, individuals are representation of the solutions in the domain. The Novelty Search paradigm proposes that nature has no objective, but just evolves, and any result of this evolution is because of natural selection. It is simply the concatenation of multiple events over time. Evolutionary and bio-inspired algorithms may miss the real solution to the problem because they do not look for novelty solutions, even if they are not always the best fitted [12].

Evolution cells illustrate what has been previously explained. In fact, if we take a fitness function with a parameter that includes reproduction, unicellular organisms are much more efficient in reproduction than multicellular organisms, even if the latter are more evolved and better adapted to the environment [13]. The problem that might arise is that the evolutionary or bio-inspired algorithm does not reach the ideal solutions because it gets stuck in a local optimum (in this case, unicellular organisms). In a mathematical analogy, there are some parts of the domain of the function that are not attractive for the algorithm, even if the global optimum is there [13], so the algorithm is stopped at a local optimum of the problem. Thus, two main approaches emerge from this reasoning trying to solve the aforementioned problem. The first one is that fitness functions must be studied thoroughly in order to avoid stopping in local optimum of the problem. This is, in fact, changing the problem to be solved. However, in real world problems, different modelling approaches can be proposed in order to arrive at the same solution, and these approaches may have different properties during the optimization process. The second one is to make some changes in evolutionary and bio-inspired algorithms in order to find the optimum of the fitness function even if it is not in an attractive region. In this work, we focus on the second approach to the problem. We are going to modify a bio-inspired algorithm with the aim at exploring the parts of the domain of the function that might be unattractive, but where the solution may be allocated. This strategy has already been tried with so-called random mutation applied to PSO (M-PSO) [14]. That is to say, with a certain probability, a particle of the algorithm can mutate somewhere in the domain of the function. However, doing this without any kind of control is like flipping a coin; there is no methodology to follow and everything is left to chance that the particle reaches a better optimum. The Novelty Search paradigm just provides a performance method. Those unexplored parts of the domain of the function are explored with a methodical search.

The traditional PSO is the base of the proposed algorithm and the concepts of Novelty Search are applied in the search of unexplored areas of the solution's space. This work evaluates the Novelty Swarm Algorithm on the CEC 2005 benchmark suite [15] researching efficiency on all three kinds of problems: unimodal, multimodal and compositional functions, and then we compare them with previous approaches. The CEC2005 has been chosen because it has been proved by several papers that is a really good test for algorithms in complex functions with a lot of local optima [16–18]. After the performance evaluation with the benchmark is done, the proposed NS algorithm is implemented to optimize a compression-ignited (CI) engine to minimize the pollutant emissions and fuel consumption.

The Internal Combustion Engines (ICE) remain as the most used powertrain in transportation [19], as jet engines for air transportation and as reciprocating ICE for marine and land propulsion systems. The interest in these machines is their high fuel efficiency, reasonable durability and moderate pollutant emissions [20]. It is estimated that the transport sector is responsible for around 20% of the total production of the green house gases emissions [21]. In order to attend the regulations of emissions, numerous efforts have been done in the engine industry, specifically in the improvement of combustion systems and in the development of auxiliary devices for after treatment. This process of pollutant emission reduction is a challenge since it should also consider the fuel consumption minimization without deteriorating the engine performance, which have an antagonist behaviour. The development of the combustion process has been one of the keys for reaching the emission targets, specially for Nitrous Oxides ($NO_x$) and particulate matter (soot). Efforts for the engine development have included experimental studies testing multiple engine setting parameters for conventional fuel [22], bio-fuels [23] or more advanced HCCI concepts [24], most of them combined with optimization tools to find the best performance of the engine [25] Besides, in the analysis of the combustion performance in CI engines, computational fluid dynamics (CFD) tools have been used in large scale to simulate the in-cylinder combustion process. In this context, CFD codes are tools commonly used to reproduce, visualize and study the influence of different parameters like the bowl geometry, injection configuration and air management on the combustion behaviour [26,27]. Moreover, these codes are used as support for optimization studies due to their capability for reproducing combustion phenomena, thus generating reliable trends for decision making in design stages. Once the model is validated against experimental data, it is possible to configure different engine settings, finding an optimal design without rebuilding the experimental hardware [28]. Current literature shows a variety of different approaches used for engine optimization, from the traditional design of experiments, to the most innovative as heuristics, meta-heuristics or neural networks [20,29–32]. The combustion chamber design phase usually is a complex and time-consuming process even with advanced supercomputers, since it depends on several input variables which are highly non-linear and with crossed interaction [33].

Therefore, the motivation of this work is to obtain an optimized design of a combustion system capable of minimizing the fuel consumption and pollutant emissions at the same time using the benefits of the proposed method Novelty Swarm (NS). The first goal is an improvement of the PSO algorithm searching unexplored areas of the search space and avoiding the premature convergence of the algorithm using Novelty Search concepts which characterizes the NS. The second objective is to implement the proposed methodology for the obtention of a combustion

chamber for a CI engine based on the reduction of fuel consumption and pollutant emissions ($NO_x$ and soot), which are a major environmental issue as it was mentioned. A CFD model of the combustion system was built to predict and analyse the performance of the combustion system and hence, select the parameters with a higher impact on the system. The proposed method is configured to reduce the computational time and it includes tools for the automatic preparation of the input parameters and geometry of the system. The input parameters correspond to geometrical variables that control the bowl shape, the number of holes in the injector, the injection pressure, the swirl number and the exhaust gas recirculation rate.

The rest of this paper is structured as follows. Section 2 presents some modifications of PSO and a brief summary of other works that have studied optimization in engines. Section 4 describes the engine characteristics and the numerical methodology formulation, as well as the CFD model validation and the optimization procedure. Section 3 introduces the Novelty Swarm and explains the experiments and results of the CEC2005 benchmark. Section 5 is devoted to the integration of the CFD model and the proposed NS algorithm. Also, fitting functions are defined. In Section 6, we present the results of applying the proposed NS algorithm to CEC2005 benchmark compared with PSO, M-PSO, CAPSO, LSHADE and jSO optimization algorithms. In Section 7 the engine optimization results are shown, followed by a discussion. Finally, Section 8 outlines the conclusions.

## 2. Related work

There are a lot of works in the literature focusing on modifications of PSO and their combinations with other metaheuristics or local search algorithms.

1. IDE-PSO [8]. Employs the idea of multigroup, separating the swarm into several subgroups based on particles' performances (emotional status) during evolution.
   - Merits: Outperformed 6 PSO variants in multimodal functions.
2. M-PSO [14]: It is an implementation of PSO that includes novelty search.
   - Demerits: The novelty search is practically random and not methodological.
   - Merits: It is a PSO novelty search algorithm.
3. HPSOGA [34]: Hybrid PSO and genetic algorithm with population partitioning.
   - Demerits: Stagnation with the increase of the evaluations. Tested on a reduced set of problems.
   - Merits: Good solutions at the beginning. Premature convergence.
4. Hybrid approach of PSO and Differential Evolution [35]: A framework is proposed based on each particle's social and cognitive experience (memory-swarm).
   - Demerits: Exploitative mutation strategies may deteriorate its performance.
   - Merits: It is a promising proposal. Incorporating this framework in any PSO variant was straightforward and significantly enhanced the performance of most of the considered cases.
5. ALC-PSO [36]: It is characterized by assigning the leader of the swarm with a growing age and a lifespan, which is adaptively tuned according to its leading power.
   - Merits: Designed to overcome the problem of premature convergence without significantly impairing the fast-converging feature of PSO.
6. SL-PSO [37]: A surrogate-assisted PSO algorithm and a surrogate-assisted social learning-based PSO (SL-PSO) algorithm work together to find the global optimum. SL-PSO explores, and both share promising solutions.
   - Merits: Find high-quality solutions for high-dimensional problems on a limited computational budget.
   - Demerits: Tested with a reduced set of problems.
7. CAPSO [38]: Proposes a centripetal accelerated PSO (CAPSO) based on PSO and Newtonian's Motion Laws.
   - Merits: Accelerates convergence of PSO. Compared to well-known PSO algorithms in the literature, CAPSO yields better results.
8. C-APSO [39]: Several chaotic maps are utilized to tune the main parameter of the accelerated PSO.
   - Merits: Very good performance in comparison with other chaotic algorithms.
   - Demerits: Reduced benchmark. Only compared with chaotic algorithms.
9. L-SHADE [40,41]: It is a history-based adaptive Differential Evolution algorithm with linear population size reduction.
   - Merits: It outperforms most of the compared methods for each benchmark set.
10. jSO [42]: With the same idea as L-SHADE with a new weighted version of the mutation strategy.
    - Merits: The jSO algorithm indicates better overall results when compared to the former L-SHADE algorithm on almost all dimensions.

Other comparative differential evolution approaches are reviewed in [43].

After this revision, we are going to select some algorithms for further comparisons. We select M-PSO, because it uses novelty search in a random way, and it can be a good touchstone for comparing our proposal. Other options are the well-tested algorithms CAPSO, L-SHADE and jSO.

Novelty Search has been used in the recent years in order to improve some optimization algorithms, such as Differential Evolution [44] or Genetic Programming applied to clustering [45] or automatic generation of game levels [43]. The fact that this paradigm has been used previously with relative success is a good sign of its capabilities.

In summary, the approaches that have been presented so far to improve PSO performance do not seem to be evaluated for all types of functions. Also, the problem of premature convergence is still an inconvenience in some of the results. In this work we propose a modification of the PSO which solves some of the problems discussed above. In order to provide this solution, we include the concepts of Novelty Search in a new version of the PSO algorithm, Novelty Swarm optimization algorithm. In addition, we apply it to an optimization problem in the field of ICE design.

The optimization of the ICE design is a complex problem since the combustion performance depends on high number of parameters that most of the times are nonlinear, with strong exponential behaviour of certain responses and with high cross-effects between variables. The consequence is that the response function would have multiple local optimum values making a

challenge to find the global optimum. Another characteristic of the optimization in engines is that the evaluation of the fitness function is highly time consuming, specially when it is coupled with CFD, therefore only few iterations and evaluations can be done. Previous studies proposed a combination of artificial neural networks and ant colony optimization algorithms for reducing soot and $NO_x$ emissions in a diesel engine considering four inputs variables [29]. Although the work showed potential for application in controlling systems of diesel engines, it is dependent on the algorithm parameters that might cause rapid convergence to local optimum values. Moreover, heuristics algorithms have shown to be effective methods in the optimization of combustion systems. Broatch et al. in [30] combined a CFD modelling with genetic algorithm (GA) technique to optimize the combustion system of a diesel engine, minimizing the indicated specific fuel consumption and combustion noise, while restricting the soot and $NO_x$. The simulations helped to understand the problem, although at a very high computational cost. Another study that implements GA techniques for engine optimization was conducted by Zubel et al. [46], where they combined CFD models and a microgenetic algorithm for the obtention of a piston bowl and injector nozzle geometry design using a fuel with high oxygen content as dimethyl ether. In their work, they tested two possible bowl configurations and they found that the injector design has an impact on the combustion performance. The combination of the spray angle and the piston profiles leaded to a reduction of HC and CO emissions and an improvement of engine efficiency. Recently, Badra et al. [47] presented an optimization of a combustion system using CFD and Machine Learning for reducing the fuel consumption and the pollutant emissions in a compression ignition engine fuelled with gasoline. In their procedure a design of experiments (DoE) optimization was performed followed by a Machine Learning − Grid Gradient Ascent approach. Their results show an improvement in the CO and soot emissions. One of the contributions of their exercise is the obtention of acceptable results with relatively low computational cost, compared against traditional CFD-DoE approaches. However, one of the aspects of that study was the narrow range of the parameters tested. In [31], Bertram et al. presented a hybrid method combining PSO and GA for engine performance optimization indicating that the hybrid method offered benefits of both algorithms while preventing the drawbacks of either method alone, such as fast convergence in basic PSO. They concluded that the PSO would provide better results with some modifications or combined with other approaches. In this paper we use the NS algorithm in the optimization problem of reduction of emissions and fuel consumption in combustion systems based on the fast rate of convergence, simple implementation, possibility to use a wide range of inputs parameters and the asynchronous nature of the algorithm what makes possible a time reduction of the optimization process.

## 3. Implementation of novelty search to the particle swarm optimization algorithm

Swarm intelligence is a computational paradigm that was introduced at the end of the decade of the 1980s, and published in the early 1990's [48]. It is based on the assumption that artificial intelligence cannot depend solely on individual behaviour and should take into account the influence of the society. A swarm is formed by an indeterminate amount of *particles* making elemental actions. All particles interact among themselves and the environment without a central control forcing the particles to do any specific action. Particles are subjected to limited abilities for problem resolution. Nevertheless, the interaction among the different particles which form the swarm allows individual behaviour to be improved, thus obtaining good solutions in many different scenarios [49].

PSO is a bio-inspired meta-heuristic algorithm developed by James Kennedy and Russell Eberhart in 1995 [1]. It is based on the social behaviour of bird flocks, which try to make their movements in the most optimal possible way to find food. Since its proposal, the PSO algorithm has been successfully applied to solve a large amount of optimization problems.

Mathematically, given a function $f$

$$f : \mathbb{R}^D \to \mathbb{R}, \tag{1}$$

the optimization problem consists of finding

$$x_{\text{opt}} | f(x_{opt}) \leq f(x), \qquad \forall x \in \mathbb{R}^D, \tag{2}$$

in the minimization case, and

$$x_{\text{opt}} | f(x_{opt}) \geq f(x), \qquad \forall x \in \mathbb{R}^D, \tag{3}$$

in the maximization case.

The $D$-dimensional domain of function $f$ in $\mathbb{R}^D$ is called the search space. Every point characterized by the $D$-dimensional vector represents a candidate solution to the problem. These vectors are called particles [50].

Let $X$ be a *swarm*, formed by $N$ particles:

$$X = [x_1, x_2, \ldots, x_i, \ldots, x_N], \tag{4}$$

where each particle is a D-dimensional vector in the domain of $f$,

$$x_i = [x_{i1}, x_{i2}, \ldots, x_{iD}]. \tag{5}$$

During the search for the optimum of the function $f$, a maximum number of iterations $T$ is set. For each iteration $t$, particles update their position according to the following formula:

$$x_i(t+1) = x_i(t) + v_i(t+1), \tag{6}$$

where the velocity vector $v_i(t+1)$ is updated according to:

$$\begin{aligned} v_i(t+1) =& w\ \beta \cdot v_i(t) + c_1\ \tau \cdot (p_i - x_i(t)) \\ &+ c_2\ \gamma \cdot (g - x_i(t)). \end{aligned} \tag{7}$$

$w$ is the inertia weight, $c_1$ is the individual weight and $c_2$ is the social weight. This three parameters are scalars and their values are settled independently from each other. $w$ values are usually in the range [0.5, 1.5], $c1$ and $c2$ are usually in the range [1, 3], and these values depend on the problem to be solved. Also, $p_i$ stands for the current best position of $x_i$ (local best position) and $g$ stands for the position with the best value among all the particles which have formed the swarm (the global best position).

$\beta$, $\tau$ and $\gamma$ are random vectors. Every element of each random vector is a different sample of a uniform distribution between 0 and 1. The component-wise product $\cdot$ between vectors is carried out as:

$$a \cdot b = (a_1 b_1 \ldots a_D b_D),$$

where $a = (a_1, \ldots, a_D)$ and $b = (b_1, \ldots, b_D)$.

A pseudo-code of the PSO algorithm is shown in Algorithm 1.

### 3.1. Implementation of novelty search in PSO

As in many biological systems such as ant colonies or beehives, swarm particles may have different roles in the interaction of the community. Because of this, we deploy two different families of particles, where each family has its own task and is ruled by a different velocity equation.

The first family is formed by "conqueror" particles, ruled by Eq. (7) as in the classical PSO. The name is given because they have to "conquer" the optimum of the problem. The second family is formed by "explorer" particles, where the Novelty Search

**Algorithm 1** Particle Swarm Optimization (PSO) — Minimization.

**Require:** $w, c_1, c_2, C, p$
**Ensure:** *Min of* $f$
1: **for** Every particle $i = 1, ..., C$ **do**
2: Initialize the position $x_i(0)$
3: Evaluate the best individual position $p_i = f(x_i(0))$
4: **end for**
5: Initialize the best global position $g = p_i$ such that $f(p_i)$ is the minimum.
6: **while** The stop criterion is not satisfied **do**
7: **for** Every particle $i = 1, ..., C$ **do**
8: Update velocity according equation (7)
9: Update position according to equation (6)
10: Evaluate the function $f(x_i(t+1))$
11: **if** $f(x_i(t+1)) \leq f(p_i)$ **then**
12: $p_i = x_i(t+1)$
13: **end if**
14: **if** $f(x_i(t+1)) \leq f(g)$ **then**
15: $g = x_i(t+1)$
16: **end if**
17: **end for**
18: **end while**

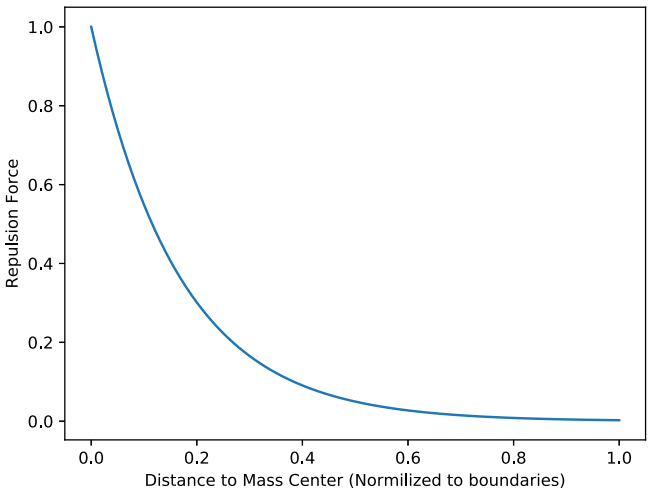



**Fig. 1.** Evolution of centre of mass repulsion in Eq. (9), $\alpha = 5$. Repulsion decreases with the distance to the centre of mass. The repulsion force is normalized to the boundaries of the domain of the function in the $x$ axis and normalized to the maximum value in the $y$ axis, avoiding to move away from the boundaries.

concept is applied. The name is given because they "explore" the unexplored/less attractive region of the search space.

In order to implement Novelty Search in the algorithm, we store all the particles generated by the algorithm in a repository, and the new explorer particles will avoid the regions close to the particles in the repository. If we call $\mathcal{R}(t)$ the repository in the iteration $t$, we define:

$$MC(t) = \frac{\sum_{x \in \mathcal{R}(t)} x}{card(\mathcal{R}(t))}, \tag{8}$$

where $card(\mathcal{R}(t))$ is the number of elements of $\mathcal{R}(t)$. $MC(t)$ is the point in the search space that represents all the points in the system in such a way that the behaviour of $MC(t)$ summarizes the behaviour of the system in the iteration $t$. By analogy on the physics idea of the centre of mass of an object, we are going to call $MC(t)$ centre of mass.

At the beginning of the algorithm, the repository of the particles is small and the computational cost of calculating the centre of mass of the system is small. However, when the repository grows as the number of iterations of the algorithm increases, the computational cost becomes expensive. To avoid this problem, the size of the repository that stores the positions of the particles that have formed the swarm must have a maximum value. This value must be big enough to represent the proper dispersion of the particles in the domain of the function, but small enough to avoid a high computational cost in the evaluation of the centre of mass. In order to obtain a statistical significance sample of the population, when $\mathcal{R}(t)$ reaches a prefixed maximum number of particles, say $p$, we select $q < p$ particles randomly as representative of all the particles in the repository. $q$ should be a number that allows a fast calculation of the centre of mass. Also, to prevent the excessive weight of local optimum where the conqueror particles may get stuck, only the explorer particles and the initial conqueror particles are added to the repository.

Thus, the equation that rules the explorer particles is a modification of Eq. (7), adding the interaction with the centre of mass and removing the interaction with the global best position of the swarm,

$$\begin{aligned} v_i(t+1) = & w\ \delta \cdot v_i(t) + c_1\ \phi \cdot (p_i - x_i(t)) + \\ & c_3\ \rho \cdot \exp\left(-\alpha \cdot \left|\frac{x_i(t) - MC(t)}{x_{max} - x_{min}}\right|\right) \cdot (x_i(t) - MC(t)), \end{aligned} \tag{9}$$

where $x_{max}, x_{min}$ are vectors of dimension $D$ that represent the boundaries of the search space and $\delta$, $\phi$ and $\rho$ are random vectors like in Eq. (7). The quotient $\frac{x_i(t)-MC(t)}{x_{max}-x_{min}}$ should be carried out componentwise.

Eq. (9) has three different elements corresponding to each of the terms on the right-hand side:

- Inertia (first term): The particle follows its previous direction.
- Individual best (second term): The particle is attracted by its best position.
- Centre of mass (third term): The particle is repulsed by the position of the centre of mass. Fig. 1 shows the decrease in repulsion as the particle moves away from the centre of mass, modulated by $\alpha > 0$.

The pseudo-code of Novelty Search implementation in PSO algorithm is shown in Algorithm 2. This new algorithm is named Novelty Swarm (NS)

Comparing Algorithms 1 (PSO) and 2 (NS), we can see:

- Algorithm NS includes the new explorer particles (lines 6–10) and its evolution is described in lines 25–36.
- The explorer particles evolve looking for unexplored parts of the parameter space. This is done via expression (9), where the evolution of the explorer particle consists of being repelled by the centre of mass (line 26).
- The conqueror particles in NS evolve as the particles in PSO, taking into account that if explorer particles obtain a better result, it is considered by the conqueror particles in their evolution.
- The conqueror and explorer particles share information in lines 21–22 and 33–34, where the global best is updated, taking into account the fitnesses of all the particles.

## 4. Combustion system design optimization

This section describes the method for the combustion system evaluation using a 3D CFD model approach. The target of the model is to predict the performance of the combustion process

**Algorithm 2** Novelty Swarm (NS) — Minimization.

**Require:** $w, c_1, c_2, c_3, C, E, p, q$
**Ensure:** *Min of* $f$
**for** Every particle $i = 1, ..., C$ (conquerors) **do**
  Initialize the position $x_i(0)$
  Evaluate the best individual position $p_i = f(x_i(0))$
  Add $x_i(0)$ to repository
**end for**
**for** Every particle $i = 1, ..., E$ (explorers) **do**
  Initialize the position $x_i(0)$
  Initialize the best individual position $p_i = f(x_i(0))$
  Add $x_i(0)$ to repository
**end for**
Initialize the best global position (shared between both families) $g = p_i$ such that $f(p_i)$ is the minimum.
**while** The stop criterion is not satisfied **do**
  Update the centre of mass according to equation (8)
  **for** Every particle $i = 1, ..., C$ (conquerors) **do**
    Update velocity according equation (7)
    Update position according to equation (6)
    Evaluate the function $f(x_i(t+1))$
    **if** $f(x_i(t+1)) \leq f(p_i)$ **then**
      $p_i = x_i(t+1)$
    **end if**
    **if** $f(x_i(t+1)) \leq f(g)$ **then**
      $g = x_i(t+1)$
    **end if**
  **end for**
  **for** Every particle $i = 1, ..., E$ (explorers) **do**
    Update velocity according to equation (9)
    Update position according to equation (6)
    Evaluate the function $f(x_i(t+1))$
    Add $x_i$ to repository
    **if** $f(x_i(t+1)) \leq f(p_i)$ **then**
      $p_i = x_i(t+1)$
    **end if**
    **if** $f(x_i(t+1)) \leq f(g)$ **then**
      $g = x_i(t+1)$
    **end if**
  **end for**
  **if** len(repository) $\geq p$ **then**
    Select randomly $q$ particles of repository
    Discard the unchosen particles from the repository
  **end if**
**end while**

through different variables, such as cylinder pressure, energy released, fuel consumption, pollutant emissions, among others. The combustion system considered in this study is described at the beginning of the section and the details of the model formulation and validation are provided subsequently. Finally, the integration of the ICE model and the optimization algorithm NS is presented.

### 4.1. Combustion system description

Fig. 2 presents a schematic diagram of a compression ignition combustion system. The combustion process starts after the injection of the fuel at high pressures, promoting its atomization and mixing with the air in the chamber. The high temperatures and pressures in the chamber cause the self-ignition of the mixture in a very short time, triggering the combustion (first premixed, and later by diffusion). A wide set of parameters affect the combustion performance, being the most relevant the geometry of the piston bowl ($p_1 - p_5$), geometry of the fuel injector nozzle ($N_o$), shape of the inlet ports (characterized by *SN*), operating conditions (such as injection pressure – *IP* – and exhaust gas recirculation rate — *EGR*), which are highly non-linear and could have crossed interaction between them. All these parameters are referred as input data in Fig. 2. The physics of the combustion process is modelled by means of state-of-the-art 3D CFD models that are able to reproduce the non-linear phenomena of the system. The target is to predict the fuel consumption (ISFC) and pollutant emissions, such as $NO_x$ and soot, in order to minimize them as much as possible (identified as output variables to minimize in the right-hand side of Fig. 2). The processes indicated in Fig. 2 correspond to the evaluation of the function of the optimization process.

### 4.2. Methods

The aim of this study is to conduct a 3D CFD-guided combustion system hardware development using efficient optimization tools to obtain an optimum combustion system through the evaluation of emissions and consumption parameters. The following is an overview of the CFD-guided design process:

1. Model formulation and validation, where the setup of the CFD model is done in order to reproduce the behaviour of the engine. Additionally, the calibration of the spray and emissions models is performed against experimental data (Section 4.3).
2. Mesh simplification, where a mesh with fewer cells than the original model is built and validated with experimental results in order to reduce the computational execution time of the model (Section 4.4).
3. Integration between the optimization algorithm and the engine model (Section 5).
4. Execution of the optimization procedure (Section 7).

### 4.3. CFD model formulation

A numerical model of the combustion chamber was developed using the open source OpenFOAM software [51]. Along with OpenFOAM, LibICE, which is a group of libraries and solvers for internal combustion engines [52], has been used. The combustion process is a combination of complex processes involving physical and chemical phenomena. Thus, in order to mimic those processes, the CFD tool couples a variety of sub-models that simulate the different mechanisms involved, for example, gas motion, spray development, chemistry and heat transmission, among others. In this study, the simulations were carried out in closed cycle, that means from the closing of the intake valves until the opening of the exhaust valves. Also, since the combustion chamber volume varies during the engine cycle, LibICE uses the dynamic mesh layering technique [53,54] to reproduce the piston movement. In order to reduce the computational cost of the model, which is important in this type of studies where hundreds of simulations are carried out, only an axisymmetric sector (1/10) of the geometry is simulated. While n-heptane was used as the diesel surrogate, a Lagrangian-typo was used to mimic the spray, assuming a "Blob" injection method [54,55]. Liquid droplets were grouped into parcels to statistically represent the spray field in the combustion chamber using a specific rate of injection (ROI) profile with a virtual injector model [56]. Parcels evolve into the CFD domain exchanging mass, momentum and energy with the continuous gas phase. Additional submodels are necessary to reproduce the liquid atomization, break-up, heat transfer and evaporation. Both, the Kelvin–Helmholtz (KH) and Rayleigh–Taylor (RT) algorithms, were adopted for the secondary break-up process [57,58]. To model the in-cylinder turbulence, the

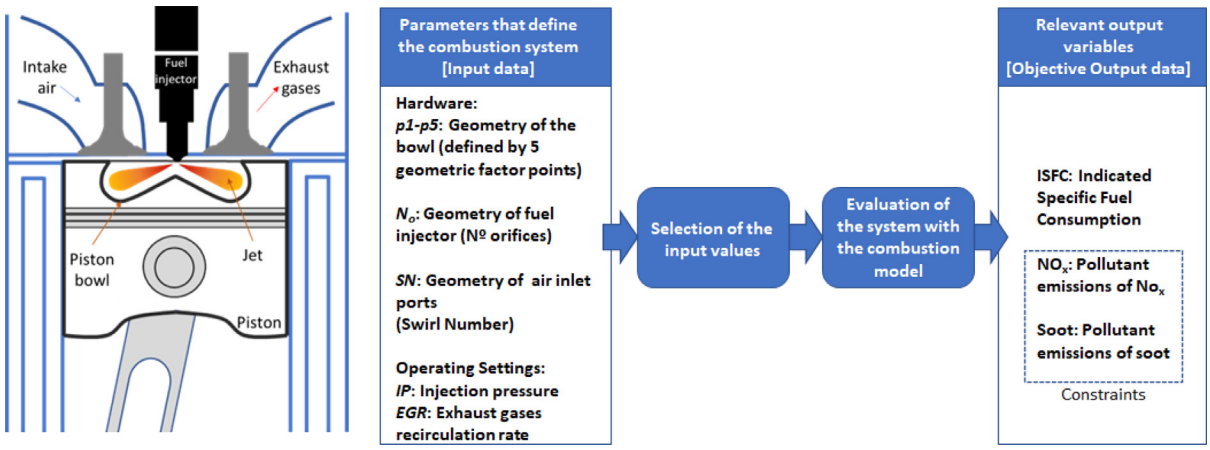



**Fig. 2.** Diagram of a combustion system. Parameters and variables considered in the modelling procedure.

**Table 1**
Models specifications.

| | |
|---|---|
| Turbulence | RNG $k-\epsilon$ RANS |
| Wall heat transfer | Angelberger |
| Spray models | Injection: Blob Injector<br>Break-up: KH-RT<br>Collision: off<br>Evaporation: standard |
| Combustion model | RIF-based tabulation |
| Chemical mechanism | NC7Curran |
| Soot model | Leung Lindstedt Jones |

Reynolds-Averaged Navier Stokes (RANS) based on re-normalized group (RNG $k-\epsilon$) was used to perform all the simulations [59]. Pressure and velocity equations are coupled by the PIMPLE algorithm. The Angelberger model was used to compute the wall heat transfer coupled with the turbulence model [60]. For fuel chemistry, a reduced chemical kinetic mechanism for primary reference fuel consisting on 162 species and 1543 reactions was used. Combustion simulations were run using the representative interactive flamelets (RIF) model, that is based on the laminar flamelet concept and assumes that the chemical scales are much smaller than the turbulent time and length scales [55]. The characteristic of this model is that reacting scalars depend on the mixture fraction variable, which is proportional to the local fuel-to-air ratio. For more detailed information about the combustion model the reader is referred to [61]. Although there are more detailed and accurate models available in the literature, the choice of the sub-models in this study was a trade-off between accuracy and computational efficiency for such a large optimization process. A list of the sub-models used is provided in Table 1.

In order to configure the cases automatically, new tools were developed for the preparation of the CFD models:

- **Geometry generator**: To generate a combustion chamber with plausible design, a piston bowl profile generator was implemented using Bézier polynomial curves with five parameters [62]. All five parameters are dimensionless and have their own ranges and limits providing more freedom in the bowl profile generation. In Fig. 3, it can be seen the five parameters and two examples of bowl profiles generated by the method. The crevice, which is a small region between piston and cylinder liner, was kept constant in shape during this study. This parametrization approach allowed a flexible variation of the bowl over a wide range. Changing the individual design parameters of the piston bowl obviously has an impact on the compression ratio (CR), which is supposed to stay fixed to a predefined value of 16 in this study. The CR was compensated changing the squish height. Besides, since the spray targeting affects the engine performance, the spray angle was kept constant in this study. The main challenge in the mesh orientation are the re-entrant geometries that can provide negative volumes when the mesh is created. After the bowl profile is defined, every mesh was generated using a python script that defines all the blocks, cells and nodes automatically. The procedure for the automatic mesh generation is described in [63]. The Bézier bowl curve generated is an input and the control points for block definition were updated according the bowl step and re-entrant curvature to avoid negative volumes and skewness issues. The control points and block definition can be seen in Fig. 4. Additionally, the number of injector orifices was an input for defining the sector mesh.
- **Virtual Injector Model**: The injection rate profile is a key factor in the combustion of CI engines, since it is able to affect the performance and emissions levels. Considering this, an in-house code was developed to automatically generate the injection profile from the combination of three parameters. They are the injection pressure, the number of injector holes and the total mass fuel injected. The profile of ROI has a trapezoidal form and the code assumes incompressible flow across the nozzle holes and apply the equations of continuity and Bernoulli between inlet and outlet of the orifices [56,64]. In this study the discharge coefficient was considered equal to 0.88 for all cases (this value was obtained from experimental measurements). Fig. 5 depicts the good agreement of the curve obtained with the Virtual Injector Model against the experimental data.

### 4.4. Model validation and mesh simplification

A first CFD model was calibrated to reproduce the experimental data. A regular production diesel engine has been used as the experimental platform to evaluate the CFD modelling performance and the current combustion conditions. The engine is a four-cylinder configuration equipped with a turbocharger. A ten-nozzle injector with hole diameter of 112 μm with a spray angle of 154° was used in all experimental tests. The engine has a compression ratio (CR) of 16 and the main specifications of the engine are in the Table 2

The engine was operated at 3700 rpm and full power. All the boundary conditions used to simulate the engine were obtained from experiments using the methodology explained in [65]. The left-hand side image in Fig. 6 presents a cross-section view of

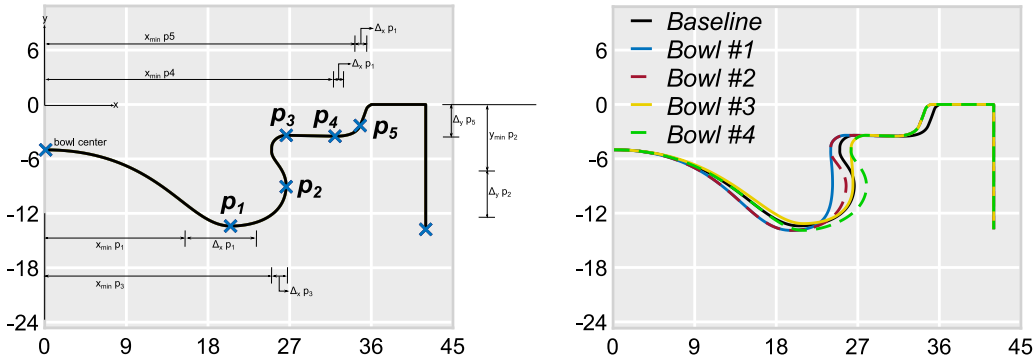



**Fig. 3.** Parameters definition for Bézier curves (Left-hand side) and example of various generated bowl profiles (Right-hand side).

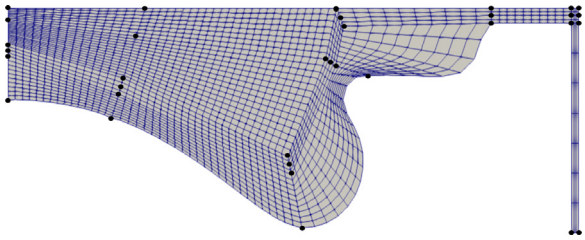


**Fig. 4.** Block definition and mesh control points location.

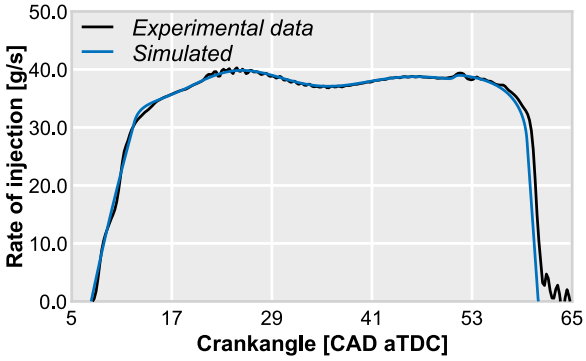



**Fig. 5.** Rate of injection comparison: experimental data versus in-house code.

**Table 2**
Engine specifications.

| | |
|---|---|
| Engine type | Direct-injection diesel engine |
| Number of cylinders [–] | 4 |
| Volume [l] | 2.2 |
| Bore – stroke [mm] | 85–96 |
| Connecting rod length [mm] | 152 |
| Compression ratio [–] | 16:1 |
| Injector number of holes | 10 |

the mesh used at top dead center (TDC). It consists of 52 000 cells at the TDC and 398 000 cells at the intake valves closing (IVC). The used mesh was generated automatically using the method developed in [53]. The mesh motion is integrated in the solver and is composed by multi-regions where each region of the mesh motion is accommodated in different ways. Also, the tool combines the use of different topological changes and polyhedral vertex-based motion solver for mesh deformation based on Finite Element Method.

In Fig. 7, the in-cylinder pressure and the heat release rate (HRR) of the simulation were compared with the experimental data. The black line refers to the experimental data, and

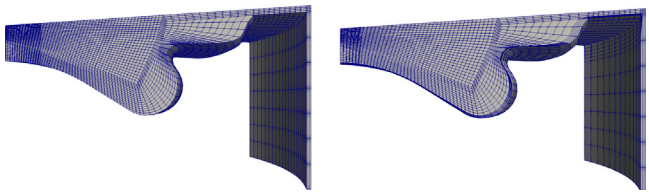


**Fig. 6.** Fine mesh (left) versus coarse mesh (right).

**Table 3**
Comparison between fine and coarse mesh.

| | Fine mesh | Coarse mesh |
|---|---|---|
| Cell count at TDC | 52 000 | 26 900 |
| Cell count at IVC | 398 000 | 203 300 |

the blue line depicts the results using this mesh (fine mesh). Based on Fig. 7, the predictions are in fairly good agreement with the experimental data and they provide confidence that the comprehensive design optimization is valid. Despite of the good agreement between the simulations and experiments, CFD simulations are highly time consuming. This compromises its suitability in the use of the optimization technique, that requires a large number of model evaluations. Then, the original model was simplified, using a coarser mesh to reduce the number of cells. In this sense, the calculation time was reduced while ensuring enough accuracy-level. The characteristics of the coarse mesh in relation to the fine mesh are presented in Table 3 and in the right-hand side image in Fig. 6.

Using the coarse mesh, the predictions also show a good agreement with the experimental data. The comparison between experimental data, fine and coarse mesh is shown in Fig. 7. As expected this configuration provides a simulation time of about 24 h (on 4 processors), while the fine mesh takes around 30 h of run-time (on 4 processors). Based on the simulation execution time, the coarse mesh configuration will be employed in the optimization process. Once the CFD simulation is performed, the results are evaluated using a fitness function in order to determine the quality of the solution.

## 5. CFD-NS algorithm integration

The process optimization is defined to find a combustion system that minimizes the fuel consumption (ISFC) while maintaining the pollutant emissions levels under the reference value ($NO_x$ and soot). The optimization algorithm should handle the high dimension of parameters, which are non-linear, a big search space, and a function evaluation that has a high execution time. Therefore, the NS algorithm has been selected (based on the

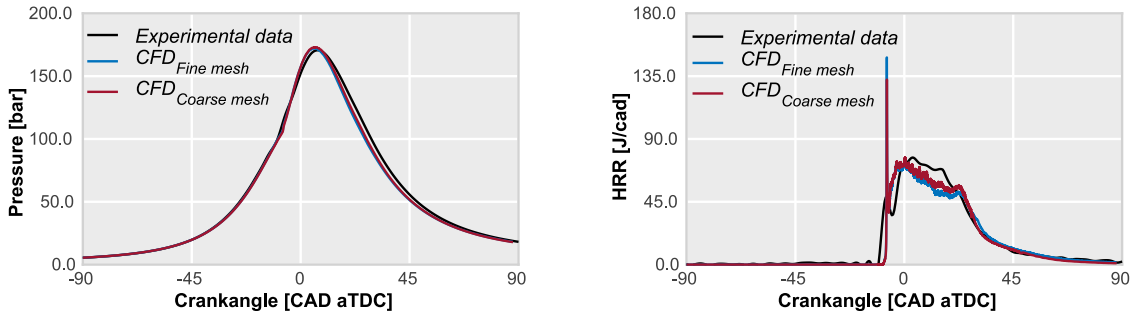



**Fig. 7.** Comparison between experimental data (black curve) and simulation results for fine and coarse meshes (blue and red curves respectively). Left side: Evolution of the in-cylinder pressure. Right side: Estimated Heat Release Rate.

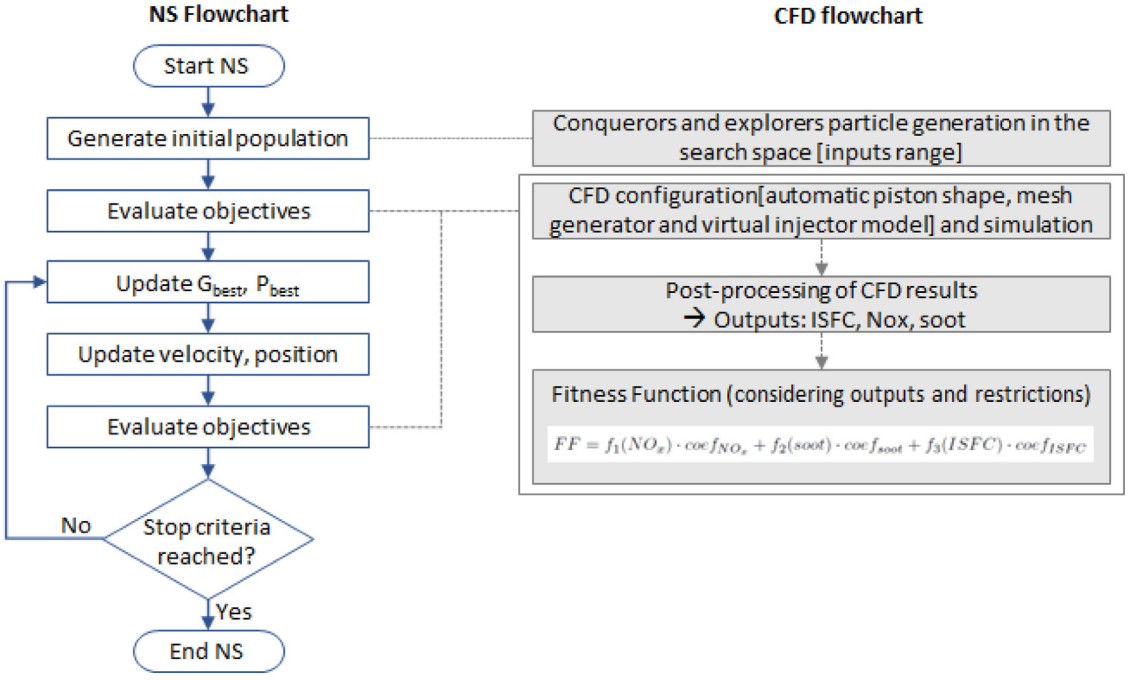



**Fig. 8.** Global methodology: Integration of the optimization method and CFD model (evaluation of the function).

benchmark results presented in next Section 6). The complete integration of the NS algorithm and the CFD model studied in this research is described in Fig. 8. The procedure for the NS algorithm and the function evaluation using CFD includes the following steps:

1. Candidate solutions are initialized inside the domain of the search space (conqueror and explorer particles).
2. Candidate solutions are evaluated to obtain the value of the fitness function. This is performed configuring the CFD model (using the input values generated by the NS) and with the CFD simulation.
3. Determination of the Output variables: ISFC, $NO_x$ and soot. Since these three outputs might have a different trend, then the overall behaviour of the system is evaluated by means of a unique Fitness Function ($FF$) that considers the three variables at the same time. The fitness function is detailed at the end of this section.
4. Optimization algorithm variables are updated ($G_best$, $P_best$, velocity, position).
5. New candidate solutions are created.
6. Go to step 2 if stop criteria is not reached.

### 5.1. Optimization parameters and fitness function

Nine parameters related to combustion system design were chosen as inputs for the optimization (given in Fig. 2). All the parameters and their ranges (Table 4) were selected considering technological limitations in the manufacture of the optimized combustion system. There are five geometrical parameters for defining diverse geometries with a certain degree of freedom ($p_1$: $p_5$ identified in Fig. 3). The others are related to the injection system as the number of holes ($N_o$) and injection pressure ($IP$), another associated with the air motion (swirl number, SN), and the last one is the exhaust gas recirculation rate (EGR). All the parameters were chosen due to their influence on the combustion process and also on the engine emissions.

The main output variable to minimize is the ISFC, while constraining the exhaust emissions of $NO_x$ and soot under the reference level and reducing them as much as possible. The objective is to ensure that the obtained system produces lower emissions than the reference case. Then, configurations that exceed the emission constrains are accordingly penalized. The values of $NO_x$ and soot selected as constraint values are the ones obtained in the CFD simulations of the reference case used for the model validation (Section 4.4).

**Table 4**
Range of the input parameters considered in the optimization.

| Parameter | Range |
|---|---|
| $p_1$: Geometrical parameter 1 [–] | [−0.5, 1.0] |
| $p_2$: Geometrical parameter 2 [–] | [−1.0, 1.25] |
| $p_3$: Geometrical parameter 3 [–] | [−1.0, 1.0] |
| $p_4$: Geometrical parameter 4 [–] | [0.0, 1.0] |
| $p_5$: Geometrical parameter 5 [–] | [−1.4, 0.1] |
| $N_o$: Number of injector nozzles [–] | [4, 12] |
| $SN$: Swirl number at IVC [–] | [1.0, 3.0] |
| $IP$: Injection pressure [bar] | [1500, 2000] |
| EGR [%] | [0, 30] |

The fitness function was formulated to consider the relative importance of ISFC, Soot and $NO_x$ in the minimization process and taking into account the constraints values. These considerations were expressed in the fitness function as:

$$f_1(NO_x) = \begin{cases} \frac{NO_{x,CFD}}{NO_{x,lim}} & if \quad NO_{x,CFD} < NO_{x,lim} \\ \frac{NO_{x,CFD}}{NO_{x,lim}} + 100 \cdot (NO_{x,CFD} - NO_{x,lim})^2 & if \quad NO_{x,CFD} \geq NO_{x,lim} \end{cases} \tag{10}$$

$$f_2(soot) = \begin{cases} \frac{-log(soot_{CFD})}{log(soot_{lim})} & if \quad soot_{CFD} < soot_{lim} \\ \frac{-log(soot_{CFD})}{log(soot_{lim})} + 1000000 \cdot (log(soot_{CFD}) - log(soot_{lim}))^2 & if \quad soot_{CFD} \geq soot_{lim} \end{cases} \tag{11}$$

$$f_3(ISFC_x) = \begin{cases} \frac{ISFC_{CFD}}{ISFC_{lim}} & if \quad ISFC_{CFD} < ISFC_{lim} \\ \frac{ISFC_{CFD}}{ISFC_{lim}} + 100 \cdot (ISFC_{CFD} - ISFC_{lim})^2 & if \quad ISFC_{CFD} \geq ISFC_{lim} \end{cases} \tag{12}$$

And the total fitness function is

$$FF = f_1(NO_x) \cdot coef_{NO_x} + f_2(soot) \cdot coef_{soot} + f_3(ISFC) \cdot coef_{ISFC} \tag{13}$$

where $NO_{x,CFD}$, $soot_{CFD}$ and $ISFC_{CFD}$ are the values obtained in the CFD simulation, whereas $NO_{x,lim}$, $soot_{lim}$ and $ISFC_{lim}$ refers to the emission levels achieved in the baseline configuration. Finally, $coef_{NO_x}$, $coef_{soot}$ and $coef_{ISFC}$ are coefficients used to balance the equation according to the order of magnitude of each term. The value of the objective is the one that feeds back the NS algorithm for the following iteration.

As the main objective is to increase the efficiency of the engine at the same time as it reduces the $NO_x$ and soot emissions, the values of the coefficients used are: $coef_{ISFC} = 50$, $coef_{NO_x} = 5$ and $coef_{soot} = 5 \cdot 10^{-5}$.

The best case of the optimization will be the one with the lowest $FF$ value

## 6. CEC2005 experimental results

In this section, first, we show the CEC2005 benchmark set, a well known and referenced set of problems for evaluation of computational intelligence problems. Then, we describe comparison among the proposed NS with PSO [50], M-PSO [14,50], CAPSO [38], LSHADE [40,41] and jSO [42] applied to the CEC2005. Later, in Section 7 we show the performance of NS algorithm on the problem of reduction of emissions in combustion systems.

**Table 5**
CEC2005 functions execution data [15]. Init. column is the region of the search space where the particles are initialized, Domain column is the search space limit, F(x*) column is the minimum value of the function and Accuracy column is the maximum error allowed to consider the problem solved.

| | Init. | Domain | F(x*) | Accuracy |
|---|---|---|---|---|
| F1 | $[-100, 100]^D$ | $[-100, 100]^D$ | −450 | −450 + 1e−6 |
| F2 | $[-100, 100]^D$ | $[-100, 100]^D$ | −450 | −450 + 1e−6 |
| F3 | $[-100, 100]^D$ | $[-100, 100]^D$ | −450 | −450 + 1e−6 |
| F4 | $[-100, 100]^D$ | $[-100, 100]^D$ | −450 | −450 + 1e−6 |
| F5 | $[-100, 100]^D$ | $[-100, 100]^D$ | −310 | −310 + 1e−6 |
| F6 | $[-100, 100]^D$ | $[-100, 100]^D$ | 390 | 390 + 1e−2 |
| F7 | $[0, 600]^D$ | $[-600, 600]^D$ | −180 | −180 + 1e−2 |
| F8 | $[-32, 32]^D$ | $[-32, 32]^D$ | −140 | −140 + 1e−2 |
| F9 | $[-5, 5]^D$ | $[-5, 5]^D$ | −330 | −330 + 1e−2 |
| F10 | $[-5, 5]^D$ | $[-5, 5]^D$ | −330 | −330 + 1e−2 |
| F11 | $[-0.5, 0.5]^D$ | $[-0.5, 0.5]^D$ | 90 | 90 + 1e−2 |
| F12 | $[-\pi, \pi]^D$ | $[-\pi, \pi]^D$ | −460 | −460 + 1e−2 |
| F13 | $[-3, 1]^D$ | $[-3, 1]^D$ | −130 | −130 + 1e−2 |
| F14 | $[-100, 100]^D$ | $[-100, 100]^D$ | −300 | −300 + 1e−2 |
| F15 | $[-5, 5]^D$ | $[-5, 5]^D$ | 120 | 120 + 1e−2 |
| F16 | $[-5, 5]^D$ | $[-5, 5]^D$ | 120 | 120 + 1e−2 |
| F17 | $[-5, 5]^D$ | $[-5, 5]^D$ | 120 | 120 + 1e−1 |
| F18 | $[-5, 5]^D$ | $[-5, 5]^D$ | 10 | 10 + 1e−1 |
| F19 | $[-5, 5]^D$ | $[-5, 5]^D$ | 10 | 10 + 1e−1 |
| F20 | $[-5, 5]^D$ | $[-5, 5]^D$ | 10 | 10 + 1e−1 |
| F21 | $[-5, 5]^D$ | $[-5, 5]^D$ | 360 | 360 + 1e−1 |
| F22 | $[-5, 5]^D$ | $[-5, 5]^D$ | 360 | 360 + 1e−1 |
| F23 | $[-5, 5]^D$ | $[-5, 5]^D$ | 360 | 360 + 1e−1 |
| F24 | $[-5, 5]^D$ | $[-5, 5]^D$ | 260 | 260 + 1e−1 |
| F25 | $[-2, 5]^D$ | $[-5, 5]^D$ | 260 | 260 + 1e−1 |

CEC2005 benchmark [15] is a set of functions with different characteristics that has been widely used to compare the performance of optimization algorithms. It is made up of 25 functions that can be divided into:

- Unimodal functions (5). F1 to F5.
- Multimodal functions (20). F5 to F25.
  - Basic functions (7). F6 to F12.
  - Expanded functions (2). F13 to F14.
  - Hybrid composition functions (11). Formed by unimodal, basic and expanded functions. F15 to F25.

The optimum is shifted from the origin in all the functions to avoid finding it in the centre of the domain. Table 5 shows the general characteristics for each function. For the execution of the test, we implemented the benchmark in Python3 [66] programming language, using Numpy [67] as dependency package. The test was executed on a PC with an Intel Xenon E5-4620 2.20 GHz and 512 GB RAM. Each of the benchmark functions was executed 25 times for each algorithm for 2, 10 and 30 dimensions. The parameter settings for each algorithm are shown in Table 6.

According to different Refs. [1,50,68,69], the values $c_1$ and $c_2$ of the canonical PSO vary depending on the problem, but usually, they are set around values $c_1 = 2$ and $c_2 = 2$. Based on these values, a sensitivity analysis was performed with the CEC2005 benchmark to find the values of the parameters which have a better performance for NS (keeping $c_1$, deleting $c_2$, and adding $c_3$), and the values obtained are shown in Table 6.

In order to select an appropriated parameter $\alpha$, a sensitivity analysis was performed varying the value from 1 to 10. We choose $\alpha = 5$. We also performed the sensitivity analysis in order to obtain the most appropriate number of particles of NS, obtaining 60 conqueror particles and 30 explorer particles.

To set the number of particles $q$ we should take randomly from the repository, we assessed the computational cost of the centre

**Table 6**
Algorithms parameters setting.

| | NS | PSO | M-PSO | CAPSO | LSHADE | jSO |
|---|---|---|---|---|---|---|
| $w$ | 0.9 | 0.9 | 0.9 | – | – | – |
| $c1$ | 2.0 | 2.0 | 2.0 | – | – | – |
| $c2$ | 2.0 | 2.0 | 2.0 | – | – | – |
| $c3$ | 2.0 | – | – | – | – | – |
| Mutation | 0.05 | – | 0.05 | – | – | – |
| Particles | 60 | 60 | 60 | 60 | – | – |
| Explorer particles | 30 | – | – | – | – | – |
| $\alpha$ | 5.0 | – | – | – | – | – |
| $NP_0$ | – | – | – | – | $18{\cdot}D$ | $18{\cdot}D$ |
| $NP_{min}$ | – | – | – | – | 4 | 4 |
| $H$ | – | – | – | – | 6 | 6 |
| $NA_g$ | – | – | – | – | $2.6 \cdot NP_g$ | $2.6 \cdot NP_g$ |
| $p$ | – | – | – | – | 0.11 | 0.11 |
| Reference | – | [50] | [14,50] | [38] | [41] | [41,42] |

of mass with 3000 to 15 000 particles. Looking for a compromise between the costing time and including as much particles as possible, we decided to set the maximum size of the repository $p = 12\,000$ and a minimum representative number of particles to calculate the mass centre $q = 10\,000$. The execution stops when the optimum is reached (with a certain error established for each function) or when the maximum function evaluations have been done (even if the optimum has not been reached). The maximum iterations are 10 000 for 2 dimensional space and 100 000 for 10 and 30 dimensional space. The ending criteria are set by the CEC2005 benchmark, see the 5th column of the Table 5.

Tables 7, 8, 9 show, for the 25 runs performed for each function, the average difference between the optimum reached by the algorithms and the real optimum of the function given in Table 5 — column F(x*). They also show the standard deviation of the error in the 25 runs of each CEC2005 function. If one of them shows a mean with value 0, it means that the problem has been solved according to the accuracy of Table 5. Fig. 9 shows the evolution of the error of the results in Table 8 (dimension 10).

Friedman two-way analysis of variances by ranks was performed according to [70], and the results of the calculations are shown in Tables 10 and 11.

According to Table 10, it can be observed that Novelty Swarm has better performance than the other algorithms in 2 and 30 dimensions. Moreover, if we focus our attention on the performance of NS in composition functions, its performance is even better, according to Table 11. In 10 dimension, LSHADE and jSO algorithms have better performance in the long run.

In 10 dimensions, Fig. 9 shows that, regarding to the fitness function when the number of function evaluations is 1000, NS is the best algorithm. NS has better results in 6 of the benchmark functions, followed by PSO, which results are the best in 3 of the benchmark functions. This fact makes NS a good option to optimize functions when the total number of possible evaluations is low.

However, in simpler functions the search space does not need to be explored so thoroughly because the optimum is easier to find. The NS algorithm does not have such good performance in these cases compared with the other algorithms.

Algorithm complexity is calculated according to [15], where $T0$ is the time used to execute a function with basic calculations (*log*, *exp* and similar), $T1$ is the time used to execute Function 3 of CEC2005 for the specified dimension for 200 000 function evaluations and $T2$ is the mean time of 5 optimization processes with the selected algorithm of the same Function 3 with 200 000 function evaluations.

Complexity calculations of Table 12 prove that the computational cost of NS is greater than the other algorithms, and it increases with the dimension of the search space, because of the computational cost of the random sampling of particles in the repository. This fact makes NS uncompetitive against PSO and M-PSO for basic functions. Nevertheless, when the difficulty of the problem to be optimized increases (for example, a problem where the execution time of the evaluation of the function is greater than the execution time of the optimization procedure or a problem where the dimension of the search space is high), NS must be considered as a really good option.

## 7. Engine optimization results

This section presents and discusses the results of the optimization performed by NS applied to the CI engine. The main barrier to the optimization process described before is its computational cost. In order to evaluate a single engine configuration (a candidate solution of the algorithm), 4 computational cores are needed, and the time consumed by this process is near 24 h. For this reason, the maximum number of function evaluations performed in the optimization process was set to 1000. A wide range of the search space must be explored, at the same time that a better optimum than the current engine configuration must be found. Also, the topology of the search space is not known, but it is possible that several local optima exist. The initial step is the evaluation of each constraint parameter separately to verify if each particle satisfy the restrictions. Fig. 10 shows the distribution of all particles regarding $NO_x$, soot and ISFC sorted by fitness function value (the worst fitness function is sorted on the left side). The graphs in the left and middle represent respectively the $NO_x$ and soot emissions with respect to their limit values. Only those configurations which accomplish the limits of all three CFD output limits have a low enough value of the fitness function. This is the reason why, even if the $NO_x$ and the soot values are below the imposed limit, if they have bad ISFC value (this is, a higher ISFC than the baseline configuration), they are placed on the left of the figure rather than other configurations with individual worse values.

In order to locate the particle that provides the best solution within the explored range, $NO_x$, soot and ISFC were contrasted in Fig. 11. In both figures the characteristic Pareto front of CI engines can be seen, showing the trade-off between $NO_x$ and soot, and $NO_x$ and ISFC. In those plots, there are several solutions that satisfy the restrictions imposed, improving the optimizing parameters. However, even if there are particles that present better results of $NO_x$, soot or ISFC, separately, the solution of the fitness function could be high, since it depends on all parameters together. For example, if one simulation provides a low value of ISFC, probably the result of $NO_x$ is higher, because these parameters have antagonistic behaviour in engines and the fitness function presents a higher value because the $NO_x$ value penalizes the solution. For this reason, the optimum solution is focused on optimizing the ISFC and soot, while maintaining the $NO_x$ within the limits.

Based on the best solution of the fitness function and the verification of the reference limits from the Pareto's front, the optimized configuration was compared with the baseline configuration. In Fig. 12, the differences between the geometries are shown, and Table 13 shows the values of the inputs for the optimized case compared to the baseline.

Table 14 shows the values of the results for the optimized case in comparison with the baseline case, where a reduction in the pollutant emissions and consumption is obtained. From the table, it can be seen that the number of holes decreases, therefore there is more space between sprays, promoting the air entrainment and avoiding the jet-to-jet interaction that enhances the combustion performance. At the same time, the injection pressure is slightly

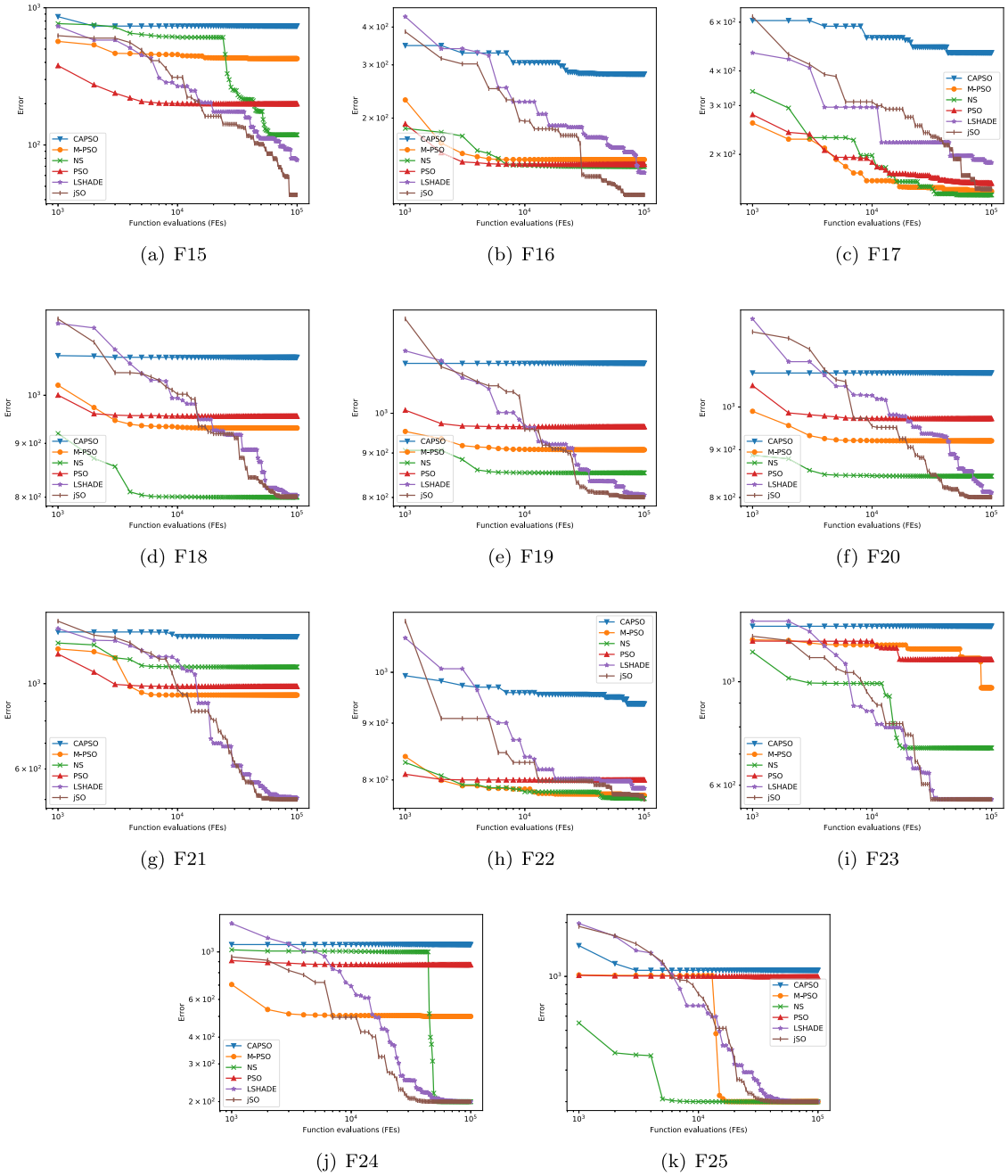
**Table 7**
CEC2005 2 dimensional optimization summary results. In bold, the best result for each function.

| | | NS | PSO | M-PSO | CAPSO | LSHADE | jSO |
|---|---|---|---|---|---|---|---|
| F1 | Mean | **0.0000E+4** | **0.0000E+4** | **0.0000E+4** | 2.5579E+0 | **0.0000E+4** | **0.0000E+4** |
| | Std. | 0.0000E+4 | 0.0000E+4 | 0.0000E+4 | 6.9464E+0 | 0.0000E+4 | 0.0000E+4 |
| | Rank | **3** | **3** | **3** | 6 | **3** | **3** |
| F2 | Mean | **0.0000E+4** | **0.0000E+4** | **0.0000E+4** | 1.8407E+1 | **0.0000E+4** | **0.0000E+4** |
| | Std. | 0.0000E+4 | 0.0000E+4 | 0.0000E+4 | 3.5627E+1 | 0.0000E+4 | 0.0000E+4 |
| | Rank | **3** | **3** | **3** | 6 | **3** | **3** |
| F3 | Mean | **0.0000E+4** | 1.9896E+2 | **0.0000E+4** | 6.0831E+3 | 6.5469E+0 | 2.9705E−7 |
| | Std. | 0.0000E+4 | 3.3626E+2 | 0.0000E+4 | 6.8497E+3 | 3.0857E+1 | 1.4552E−6 |
| | Rank | **1.5** | 5 | **1.5** | 6 | 4 | 3 |
| F4 | Mean | **0.0000E+4** | **0.0000E+4** | **0.0000E+4** | 1.5679E+1 | **0.0000E+4** | **0.0000E+4** |
| | Std. | 0.0000E+4 | 0.0000E+4 | 0.0000E+4 | 2.1033E+1 | 0.0000E+4 | 0.0000E+4 |
| | Rank | **3** | **3** | **3** | 6 | **3** | **3** |
| F5 | Mean | **0.0000E+4** | **0.0000E+4** | **0.0000E+4** | **0.0000E+4** | 3.9825E−1 | 4.0892E−1 |
| | Std. | 0.0000E+4 | 0.0000E+4 | 0.0000E+4 | 0.0000E+4 | 1.4415E+0 | 9.0767E−1 |
| | Rank | **2.5** | **2.5** | **2.5** | **2.5** | 5 | 6 |
| F6 | Mean | 1.5353E−1 | 1.9670E+0 | **1.5905E−3** | 2.5597E+2 | 6.0756E−1 | 6.3876E−2 |
| | Std. | 3.7914E−1 | 6.5648E+0 | 7.7917E−3 | 1.0380E+3 | 1.7654E+0 | 1.1562E−1 |
| | Rank | 3 | 5 | **1** | 6 | 4 | 2 |
| F7 | Mean | 1.1201E−2 | **0.0000E+4** | 5.8820E−3 | 7.2155E−1 | 1.0299E−2 | 2.1723E−3 |
| | Std. | 1.2543E−2 | 0.0000E+4 | 1.1008E−2 | 1.0722E+0 | 3.7010E−2 | 6.9524E−3 |
| | Rank | 5 | **1** | 3 | 6 | 4 | 2 |
| F8 | Mean | **0.0000E+4** | 1.6800E+1 | 1.5201E+1 | 1.3641E+1 | 9.5117E+0 | 8.5634E+0 |
| | Std. | 0.0000E+4 | 7.3322E+0 | 8.5420E+0 | 9.0261E+0 | 8.3903E+0 | 8.4227E+0 |
| | Rank | **1** | 6 | 5 | 4 | 3 | 2 |
| F9 | Mean | **0.0000E+4** | **0.0000E+4** | **0.0000E+4** | 1.0269E+0 | **0.0000E+4** | 1.1673E−3 |
| | Std. | 0.0000E+4 | 0.0000E+4 | 0.0000E+4 | 1.7118E+0 | 0.0000E+4 | 5.7186E−3 |
| | Rank | **2.5** | **2.5** | **2.5** | 6 | **2.5** | 5 |
| F10 | Mean | **0.0000E+4** | **0.0000E+4** | **0.0000E+4** | 8.4527E−1 | 1.2980E−2 | 8.8985E−3 |
| | Std. | 0.0000E+4 | 0.0000E+4 | 0.0000E+4 | 1.1075E+0 | 3.9250E−2 | 2.3352E−2 |
| | Rank | **2** | **2** | **2** | 6 | 5 | 4 |
| F11 | Mean | 1.0246E−2 | **0.0000E+4** | 1.7311E−3 | 3.4177E−1 | 4.2881E−2 | 4.9214E−2 |
| | Std. | 9.1898E−3 | 0.0000E+4 | 4.7977E−3 | 3.4729E−1 | 6.4931E−2 | 6.5874E−2 |
| | Rank | 3 | **1** | 2 | 6 | 4 | 5 |
| F12 | Mean | **0.0000E+4** | **0.0000E+4** | **0.0000E+4** | 3.3646E+1 | 4.1021E−3 | 5.4803E−2 |
| | Std. | 0.0000E+4 | 0.0000E+4 | 0.0000E+4 | 1.2149E+2 | 1.9954E−2 | 1.3780E−1 |
| | Rank | **2** | **2** | **2** | 6 | 4 | 5 |
| F13 | Mean | **0.0000E+4** | 4.7350E−3 | 2.3676E−3 | 1.5768E−2 | 9.6409E−4 | **0.0000E+4** |
| | Std. | 0.0000E+4 | 8.4261E−3 | 6.4116E−3 | 2.3694E−2 | 4.7225E−3 | 0.0000E+4 |
| | Rank | **1.5** | 5 | 4 | 6 | 3 | **1.5** |
| F14 | Mean | 1.6945E−2 | **1.2436E−2** | 1.6907E−2 | 1.3414E−1 | 1.5054E−2 | 1.7313E−2 |
| | Std. | 6.3100E−3 | 9.3273E−3 | 6.2886E−3 | 2.3249E−1 | 6.6922E−3 | 1.8965E−2 |
| | Rank | 4 | **1** | 3 | 6 | 2 | 5 |
| F15 | Mean | **0.0000E+4** | **0.0000E+4** | 4.0000E+0 | 8.6370E+1 | 1.1218E−4 | 7.0381E−1 |
| | Std. | 0.0000E+4 | 0.0000E+4 | 1.9596E+1 | 9.0751E+1 | 5.4809E−4 | 2.6568E+0 |
| | Rank | **1.5** | **1.5** | 5 | 6 | 3 | 4 |
| F16 | Mean | **0.0000E+4** | 1.2000E+1 | 1.6000E+1 | 1.4856E+2 | 1.6090E+1 | 5.8573E+0 |
| | Std. | 0.0000E+4 | 3.2496E+1 | 3.6661E+1 | 1.1851E+2 | 3.6622E+1 | 2.1000E+1 |
| | Rank | **1** | 3 | 4 | 6 | 5 | 2 |
| F17 | Mean | 1.6326E+0 | 5.6130E+0 | 3.6200E+1 | 1.9686E+2 | **7.6461E−1** | 6.5283E+0 |
| | Std. | 7.9981E+0 | 2.0824E+1 | 4.8267E+1 | 1.5600E+2 | 2.0970E+0 | 2.1277E+1 |
| | Rank | 2 | 3 | 5 | 6 | **1** | 4 |
| F18 | Mean | **2.8153E+1** | 2.6000E+2 | 6.4647E+1 | 3.5923E+2 | 1.7270E+2 | 2.5724E+2 |
| | Std. | 7.2186E+1 | 1.6248E+2 | 9.3566E+1 | 1.6928E+2 | 1.1974E+2 | 1.0756E+2 |
| | Rank | **1** | 5 | 2 | 6 | 3 | 4 |
| F19 | Mean | **2.1116E+2** | 2.8000E+2 | 2.3438E+2 | 3.6917E+2 | 2.3727E+2 | 2.6422E+2 |
| | Std. | 4.1428E+1 | 1.2000E+2 | 7.2576E+1 | 1.1153E+2 | 7.8307E+1 | 7.4131E+1 |
| | Rank | **1** | 5 | 2 | 6 | 3 | 4 |
| F20 | Mean | **0.0000E+4** | 2.4800E+2 | 2.2405E+2 | 4.1788E+2 | 2.4197E+2 | 2.2175E+2 |
| | Std. | 0.0000E+4 | 1.1356E+2 | 7.0804E+1 | 1.7039E+2 | 8.9505E+1 | 1.0374E+2 |
| | Rank | **1** | 5 | 3 | 6 | 4 | 2 |
| F21 | Mean | **5.7223E+1** | 2.7355E+2 | 2.2420E+2 | 3.9920E+2 | 2.3957E+2 | 2.1519E+2 |
| | Std. | 8.9159E+1 | 1.6514E+2 | 1.4079E+2 | 1.5854E+2 | 1.6811E+2 | 1.5009E+2 |
| | Rank | **1** | 5 | 3 | 6 | 4 | 2 |
| F22 | Mean | **1.7321E+2** | 3.1288E+2 | 2.2414E+2 | 3.5004E+2 | 2.3841E+2 | 2.1022E+2 |
| | Std. | 7.1371E+1 | 1.2680E+2 | 8.1017E+1 | 1.1849E+2 | 8.5961E+1 | 3.7995E+1 |
| | Rank | **1** | 5 | 3 | 6 | 4 | 2 |

(*continued on next page*)

**Table 7** (*continued*).

| | | NS | PSO | M-PSO | CAPSO | LSHADE | jSO |
|---|---|---|---|---|---|---|---|
| F23 | Mean | 1.7269E+2 | 2.6094E+2 | **1.4534E+2** | 3.9930E+2 | 2.6942E+2 | 3.5543E+2 |
| | Std. | 1.1838E+2 | 2.2004E+2 | 2.2679E+2 | 2.1432E+2 | 2.0487E+2 | 1.7943E+2 |
| | Rank | 2 | 3 | **1** | 6 | 5 | 4 |
| F24 | Mean | **1.9631E+2** | 2.0000E+2 | 2.0000E+2 | 2.7823E+2 | 1.9822E+2 | 2.0000E+2 |
| | Std. | 1.0595E+1 | 0.0000E+4 | 1.1441E−6 | 1.4708E+2 | 8.7166E+0 | 9.5344E−4 |
| | Rank | **1** | 4 | 4 | 6 | 2 | 4 |
| F25 | Mean | 1.9391E+2 | 1.9825E+2 | **1.6610E+2** | 3.2487E+2 | 3.4097E+2 | 3.9252E+2 |
| | Std. | 1.8537E+1 | 1.6979E+2 | 3.5262E+1 | 2.3101E+2 | 1.6175E+2 | 1.5745E+2 |
| | Rank | 2 | 3 | **1** | 4 | 5 | 6 |




(a) F15 (b) F16 (c) F17

(d) F18 (e) F19 (f) F20

(g) F21 (h) F22 (i) F23

(j) F24 (k) F25

**Fig. 9.** Convergence of the different algorithms for composition 10 dimensional benchmark functions. The median error run is shown for each algorithm. Figures represent *Error* on the *Y* axis vs. number of function evaluations, *FE*, on the *X* axis. Results are in agreement with figures collected in Table 8.

increased, which is related to a higher spray momentum and better atomization and evaporation. Moreover, the optimized case uses an EGR ratio of 14.57%, which is a well known and effective practice for reducing $NO_x$ emissions. All these trends corroborate that the methodology is providing reasonable results, delivering a solution that is in agreement with the performance

**Table 8**
CEC2005 10 dimensional optimization summary results. In bold, the best result for each function.

| | | NS | PSO | M-PSO | CAPSO | LSHADE | jSO |
|---|---|---|---|---|---|---|---|
| F1 | Mean | **0.0000E+4** | 4.1303E+0 | **0.0000E+4** | 3.1124E+3 | 9.6519E−5 | **0.0000E+4** |
| | Std. | 0.0000E+4 | 2.0234E+1 | 0.0000E+4 | 2.1163E+3 | 5.7582E−5 | 0.0000E+4 |
| | Rank | **2** | 5 | **2** | 6 | 4 | **2** |
| F2 | Mean | **0.0000E+4** | 4.4640E+0 | **0.0000E+4** | 6.3584E+3 | 3.0075E+1 | 1.1866E−1 |
| | Std. | 0.0000E+4 | 1.5138E+1 | 0.0000E+4 | 6.1666E+3 | 1.6061E+1 | 3.4815E−1 |
| | Rank | **1.5** | 4 | **1.5** | 6 | 5 | 3 |
| F3 | Mean | **0.0000E+4** | 1.5614E+5 | **0.0000E+4** | 3.2726E+7 | 6.0690E+5 | 3.2170E+5 |
| | Std. | 0.0000E+4 | 1.4563E+5 | 0.0000E+4 | 4.9593E+7 | 3.0767E+5 | 1.9484E+5 |
| | Rank | **1.5** | 3 | **1.5** | 6 | 5 | 4 |
| F4 | Mean | **0.0000E+4** | 1.2605E+1 | **0.0000E+4** | 7.9207E+3 | 5.9575E+1 | 1.7982E+0 |
| | Std. | 0.0000E+4 | 4.1740E+1 | 0.0000E+4 | 7.8746E+3 | 2.9210E+1 | 2.4437E+0 |
| | Rank | **1.5** | 4 | **1.5** | 6 | 5 | 3 |
| F5 | Mean | 2.7818E−5 | 1.7587E+2 | **8.8682E−6** | 2.6710E+3 | 1.1128E+1 | 1.1213E−4 |
| | Std. | 2.9581E−5 | 8.6157E+2 | 1.3840E−5 | 3.1453E+3 | 4.3352E+0 | 1.8152E−4 |
| | Rank | 2 | 5 | **1** | 6 | 4 | 3 |
| F6 | Mean | 6.9881E+1 | 3.1741E+4 | 3.6373E+1 | 7.9056E+8 | 4.0620E+1 | 2.0497E+1 |
| | Std. | 1.9782E+2 | 1.5538E+5 | 6.5570E+1 | 1.3360E+9 | 2.4400E+1 | **2.3982E+1** |
| | Rank | 4 | 5 | 2 | 6 | 3 | **1** |
| F7 | Mean | 6.9383E−1 | 1.2022E+0 | **6.2975E−1** | 2.9133E+2 | 7.8953E−1 | 1.8334E−1 |
| | Std. | 3.7641E−1 | 2.3816E+0 | 4.1380E−1 | 2.2710E+2 | 1.0782E−1 | **1.3599E−1** |
| | Rank | 3 | 5 | 2 | 6 | 4 | **1** |
| F8 | Mean | 2.0343E+1 | **2.0321E+1** | 2.0366E+1 | 2.0366E+1 | 2.0373E+1 | 2.0307E+1 |
| | Std. | 6.9352E−2 | 7.2808E−2 | 7.5030E−2 | 6.2763E−2 | 6.4728E−2 | **8.8295E−2** |
| | Rank | 3 | 2 | 4.5 | 4.5 | 6 | **1** |
| F9 | Mean | 1.2474E+0 | 3.0645E+0 | 4.8101E−1 | 7.8447E+1 | **3.2076E−2** | 4.6175E−1 |
| | Std. | 1.0964E+0 | 1.3473E+0 | 5.7998E−1 | 2.0866E+1 | 1.2507E−2 | 8.3331E−1 |
| | Rank | 4 | 5 | 3 | 6 | **1** | 2 |
| F10 | Mean | 2.0656E+1 | 1.8708E+1 | 2.0584E+1 | 8.8957E+1 | 1.2226E+1 | **1.0819E+1** |
| | Std. | 1.0733E+1 | 6.9397E+0 | 1.1610E+1 | 2.9111E+1 | 2.8072E+0 | 4.7675E+0 |
| | Rank | 5 | 3 | 4 | 6 | 2 | **1** |
| F11 | Mean | 4.9123E+0 | **4.1064E+0** | 4.2574E+0 | 8.9119E+0 | 5.2369E+0 | 4.6904E+0 |
| | Std. | 1.7328E+0 | 1.4138E+0 | 1.5997E+0 | 1.8131E+0 | 7.8148E−1 | 7.5276E−1 |
| | Rank | 4 | **1** | 2 | 6 | 5 | 3 |
| F12 | Mean | 1.7229E+3 | 1.7145E+3 | 1.4790E+3 | 4.3008E+4 | **2.3047E+2** | 2.4477E+2 |
| | Std. | 3.0218E+3 | 3.6642E+3 | 4.0989E+3 | 2.6943E+4 | 1.0950E+2 | 3.2379E+2 |
| | Rank | 5 | 4 | 3 | 6 | **1** | 2 |
| F13 | Mean | 7.4303E−1 | 6.1004E−1 | 7.1509E−1 | 1.2388E+1 | 4.4500E−1 | **3.8843E−1** |
| | Std. | 2.1967E−1 | 2.4371E−1 | 2.7171E−1 | 6.2551E+0 | 1.0604E−1 | 1.7632E−1 |
| | Rank | 5 | 3 | 4 | 6 | 2 | **1** |
| F14 | Mean | 3.2538E+0 | **2.9641E+0** | 3.0484E+0 | 4.0012E+0 | 3.2106E+0 | 3.0844E+0 |
| | Std. | 3.5841E−1 | 4.9332E−1 | 5.0773E−1 | 3.7659E−1 | 2.1339E−1 | 3.3986E−1 |
| | Rank | 5 | **1** | 3 | 6 | 4 | 2 |
| F15 | Mean | 2.2687E+2 | 2.4113E+2 | 3.3183E+2 | 7.2965E+2 | 7.2330E+1 | **4.4917E+1** |
| | Std. | 2.0275E+2 | 1.6404E+2 | 2.1352E+2 | 9.6639E+1 | 2.2714E+1 | 3.1836E+1 |
| | Rank | 3 | 4 | 5 | 6 | 2 | **1** |
| F16 | Mean | 1.4210E+2 | 1.5240E+2 | 1.4930E+2 | 3.0563E+2 | 1.3508E+2 | **1.1215E+2** |
| | Std. | 2.0791E+1 | 5.6650E+1 | 2.2017E+1 | 9.7846E+1 | 1.6558E+1 | 8.5614E+0 |
| | Rank | 3 | 5 | 4 | 6 | 2 | **1** |
| F17 | Mean | **1.4974E+2** | 1.5656E+2 | 1.5669E+2 | 4.9375E+2 | 1.8725E+2 | 1.5320E+2 |
| | Std. | 2.6083E+1 | 2.9624E+1 | 3.3452E+1 | 2.2330E+2 | 1.2608E+1 | 2.1354E+1 |
| | Rank | **1** | 3 | 4 | 6 | 5 | 2 |
| F18 | Mean | 8.2300E+2 | 9.3370E+2 | 8.3822E+2 | 1.0813E+3 | 7.8654E+2 | **7.5264E+2** |
| | Std. | 1.6717E+2 | 9.8799E+1 | 1.9811E+2 | 8.8321E+1 | 7.0689E+1 | 1.2301E+2 |
| | Rank | 3 | 5 | 4 | 6 | 2 | **1** |
| F19 | Mean | 8.6158E+2 | 9.3526E+2 | 8.3661E+2 | 1.1318E+3 | 7.9574E+2 | **7.6995E+2** |
| | Std. | 1.5852E+2 | 9.8302E+1 | 1.7975E+2 | 7.6381E+1 | 4.5852E+1 | 7.7475E+1 |
| | Rank | 4 | 5 | 3 | 6 | 2 | **1** |
| F20 | Mean | 8.3881E+2 | 9.1932E+2 | 8.6334E+2 | 1.1035E+3 | 8.0170E+2 | **7.7841E+2** |
| | Std. | 1.5090E+2 | 1.2376E+2 | 1.7808E+2 | 9.7235E+1 | 4.9105E+1 | 7.2545E+1 |
| | Rank | 3 | 5 | 4 | 6 | 2 | **1** |
| F21 | Mean | 9.1765E+2 | 8.4462E+2 | 8.2229E+2 | 1.3034E+3 | 4.9655E+2 | **4.8418E+2** |
| | Std. | 3.3248E+2 | 3.7132E+2 | 3.6434E+2 | 1.8055E+2 | 5.3891E+1 | 1.1363E+2 |
| | Rank | 5 | 4 | 3 | 6 | 2 | **1** |
| F22 | Mean | 7.8531E+2 | 8.0734E+2 | 7.9384E+2 | 9.5630E+2 | 7.8631E+2 | **7.6993E+2** |
| | Std. | 4.0272E+1 | 4.9921E+1 | 4.5640E+1 | 6.4496E+1 | 5.9842E+0 | 9.7150E+0 |
| | Rank | 2 | 5 | 4 | 6 | 3 | **1** |

(*continued on next page*)

**Table 8** (*continued*).

| | | NS | PSO | M-PSO | CAPSO | LSHADE | jSO |
|---|---|---|---|---|---|---|---|
| F23 | Mean | 8.9422E+2 | 1.0661E+3 | 9.2747E+2 | 1.3079E+3 | 5.7309E+2 | **5.4337E+2** |
| | Std. | 2.0851E+2 | 1.8120E+2 | 2.4164E+2 | 5.6407E+1 | 5.3806E+1 | 4.3649E+1 |
| | Rank | 3 | 5 | 4 | 6 | 2 | **1** |
| F24 | Mean | 3.2800E+2 | 6.9416E+2 | 4.0800E+2 | 1.1351E+3 | 2.0003E+2 | **2.0000E+2** |
| | Std. | 2.1075E+2 | 2.9875E+2 | 2.1151E+2 | 1.3570E+2 | 3.5264E−2 | 0.0000E+4 |
| | Rank | 3 | 5 | 4 | 6 | 2 | **1** |
| F25 | Mean | 2.9200E+2 | 8.7564E+2 | 3.4800E+2 | 1.1496E+3 | **2.0000E+2** | **2.0000E+2** |
| | Std. | 2.0380E+2 | 2.3412E+2 | 1.8787E+2 | 1.5780E+2 | 2.2190E−3 | 0.0000E+4 |
| | Rank | 3 | 4 | 5 | 6 | **1.5** | **1.5** |

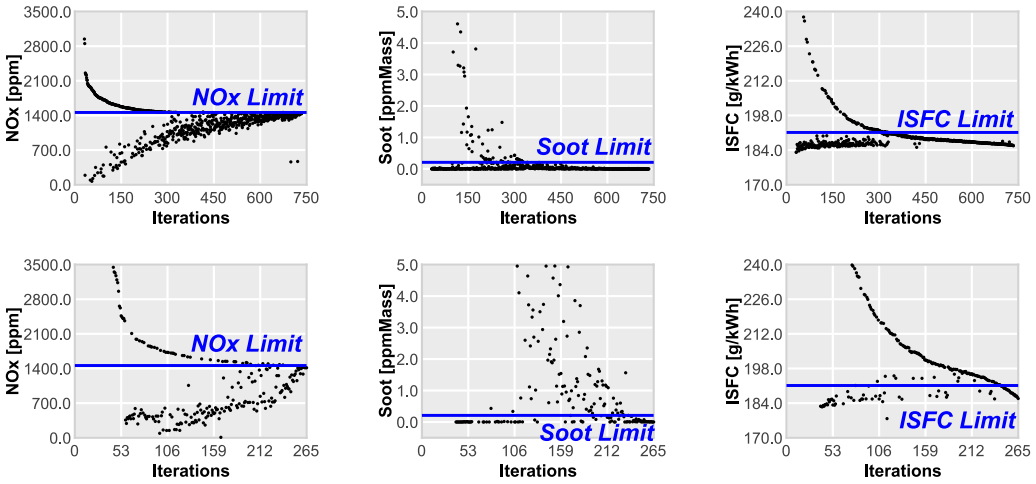



**Fig. 10.** Fitness results and limits of all particles. Upper row, conqueror particles. Lower row, explorer particles.

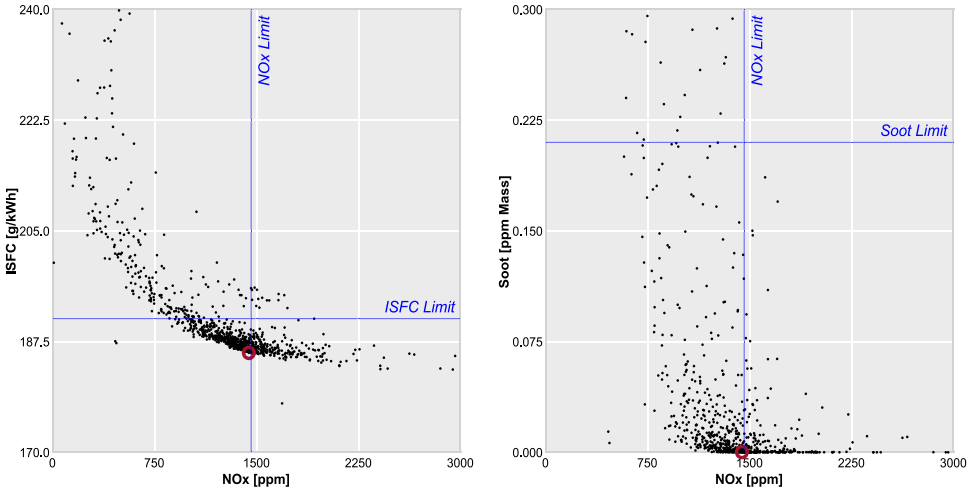



**Fig. 11.** Pareto's front of ISFC vs. $NO_x$ emissions (left) and soot vs. $NO_x$ emissions (right).

of combustion systems reported in the literature with the added value of finding the solution in less time than other methods [30].

In Fig. 13, comparison of in-cylinder pressure and rate of heat release traces between the baseline and optimized design is presented. It can be seen that the in-cylinder pressure is similar to the baseline case. However, differences in the HRR are more evident. While the premixed peak is reduced (helping to reduce combustion noise), the burning rate is increased during the non-premixed combustion phase. This could promote a higher temperature of the flame but in the limit to not generate more $NO_x$ and providing some thermodynamic advantages.

To better understand these trends, Fig. 14 shows the temporal evolution of the cylinder mass over three relevant equivalence ratios for the baseline and the optimum case. Specifically, equivalence ratio is into three different bands bounded by 0.55, 1.05 and 1.75. It can be seen that the optimized case configuration increases the mixing rate during the non-premixed combustion, subsequently rising the burning velocity (note that near stoichiometric mixtures completely disappear after 80 CAD aTDC). In contrast, the baseline case is not able to burn all the fuel during the combustion process, keeping some stoichiometric mixture in the cylinder at the exhaust valves opening. This mixing improvement leads to an enhanced combustion, that reduces soot and

**Table 9**
CEC2005 30 dimensional optimization summary results. In bold, the best result for each function.

| | | NS | PSO | M-PSO | CAPSO | LSHADE | jSO |
|---|---|---|---|---|---|---|---|
| F1 | Mean | **0.0000E+4** | 5.4503E+2 | **0.0000E+4** | 5.9236E+4 | 9.2824E+2 | 5.4566E+2 |
| | Std. | 0.0000E+4 | 5.1217E+2 | 0.0000E+4 | 2.9664E+4 | 1.5182E+2 | 8.4523E+1 |
| | Rank | **1.5** | 3 | **1.5** | 6 | 5 | 4 |
| F2 | Mean | **0.0000E+4** | 7.5932E+2 | **0.0000E+4** | 1.0966E+5 | 2.6828E+4 | 2.0175E+4 |
| | Std. | 0.0000E+4 | 1.7978E+3 | 0.0000E+4 | 4.9057E+4 | 3.9046E+3 | 3.0837E+3 |
| | Rank | **1.5** | 3 | **1.5** | 6 | 5 | 4 |
| F3 | Mean | **0.0000E+4** | 6.8772E+6 | **0.0000E+4** | 4.6978E+8 | 6.1113E+7 | 4.6188E+7 |
| | Std. | 0.0000E+4 | 3.7105E+6 | 0.0000E+4 | 4.1861E+8 | 1.4515E+7 | 9.8415E+6 |
| | Rank | **1.5** | 3 | **1.5** | 6 | 5 | 4 |
| F4 | Mean | **0.0000E+4** | 1.8306E+3 | **0.0000E+4** | 1.0738E+5 | 3.5344E+4 | 3.1080E+4 |
| | Std. | 0.0000E+4 | 1.5752E+3 | 0.0000E+4 | 4.4615E+4 | 4.0673E+3 | 3.6550E+3 |
| | Rank | **1.5** | 3 | **1.5** | 6 | 5 | 4 |
| F5 | Mean | **4.7358E+3** | 5.7386E+3 | 5.0103E+3 | 2.7579E+4 | 8.5086E+3 | 7.0942E+3 |
| | Std. | 1.4503E+3 | 1.6274E+3 | 1.9614E+3 | 5.3745E+3 | 5.1043E+2 | 6.0918E+2 |
| | Rank | **1** | 3 | 2 | 6 | 5 | 4 |
| F6 | Mean | 7.8726E+4 | 2.8159E+7 | **6.8883E+4** | 3.3743E+10 | 1.0424E+7 | 4.1943E+6 |
| | Std. | 2.5561E+5 | 4.1138E+7 | 2.5510E+5 | 3.2102E+10 | 3.2010E+6 | 8.9572E+5 |
| | Rank | 2 | 5 | **1** | 6 | 4 | 3 |
| F7 | Mean | 1.2276E+1 | 6.3930E+2 | **9.7714E+0** | 1.7625E+3 | 1.0267E+2 | 6.2657E+1 |
| | Std. | 3.8350E+0 | 5.1943E+2 | 4.5088E+0 | 7.1474E+2 | 1.7903E+1 | 8.4695E+0 |
| | Rank | 2 | 5 | **1** | 6 | 4 | 3 |
| F8 | Mean | 2.0996E+1 | **2.0962E+1** | 2.0965E+1 | 2.1008E+1 | 2.0975E+1 | 2.0986E+1 |
| | Std. | 5.8077E−2 | 7.0010E−2 | 5.9456E−2 | 4.3315E−2 | 7.2476E−2 | 5.5453E−2 |
| | Rank | 5 | **1** | 2 | 6 | 3 | 4 |
| F9 | Mean | 4.8861E+1 | 3.6908E+1 | **3.2785E+1** | 4.4602E+2 | 7.5165E+1 | 7.1191E+1 |
| | Std. | 1.1648E+1 | 1.2157E+1 | 6.0981E+0 | 6.5390E+1 | 5.1428E+0 | 4.3361E+0 |
| | Rank | 3 | 2 | **1** | 6 | 5 | 4 |
| F10 | Mean | 1.2722E+2 | **9.5479E+1** | 1.2355E+2 | 5.8149E+2 | 2.5411E+2 | 2.4166E+2 |
| | Std. | 3.6738E+1 | 3.8076E+1 | 3.8082E+1 | 1.2665E+2 | 1.6147E+1 | 1.5474E+1 |
| | Rank | 3 | **1** | 2 | 6 | 5 | 4 |
| F11 | Mean | 2.5565E+1 | **2.2611E+1** | 2.2823E+1 | 3.8885E+1 | 3.3810E+1 | 3.2043E+1 |
| | Std. | 3.5215E+0 | 4.0814E+0 | 3.5573E+0 | 3.8102E+0 | 1.5772E+0 | 1.3321E+0 |
| | Rank | 3 | **1** | 2 | 6 | 5 | 4 |
| F12 | Mean | **2.7810E+4** | 3.1153E+4 | 2.8776E+4 | 1.1503E+6 | 1.8528E+5 | 1.4218E+5 |
| | Std. | 1.7805E+4 | 2.3850E+4 | 2.2844E+4 | 4.8001E+5 | 2.1565E+4 | 1.8034E+4 |
| | Rank | **1** | 3 | 2 | 6 | 5 | 4 |
| F13 | Mean | 4.4971E+0 | **3.2139E+0** | 4.5506E+0 | 4.7560E+2 | 1.3153E+1 | 1.1606E+1 |
| | Std. | 1.0547E+0 | 8.2833E−1 | 1.2557E+0 | 3.8171E+2 | 1.2417E+0 | 1.0598E+0 |
| | Rank | 2 | **1** | 3 | 6 | 5 | 4 |
| F14 | Mean | 1.2944E+1 | **1.2734E+1** | 1.2762E+1 | 1.3525E+1 | 1.3350E+1 | 1.3289E+1 |
| | Std. | 2.7132E−1 | 3.7862E−1 | 2.9523E−1 | 2.8549E−1 | 1.4075E−1 | 1.7304E−1 |
| | Rank | 3 | **1** | 2 | 6 | 5 | 4 |
| F15 | Mean | 4.2850E+2 | 3.9838E+2 | 4.7232E+2 | 9.7835E+2 | 4.0329E+2 | **3.5594E+2** |
| | Std. | 6.3585E+1 | 1.4673E+2 | 1.8866E+2 | 1.6593E+2 | 4.2216E+1 | 4.2931E+1 |
| | Rank | 4 | 2 | 5 | 6 | 3 | **1** |
| F16 | Mean | **2.7335E+2** | 3.7899E+2 | 3.1227E+2 | 7.9408E+2 | 2.9074E+2 | 2.8187E+2 |
| | Std. | 1.4808E+2 | 1.8283E+2 | 1.4777E+2 | 1.7735E+2 | 1.7166E+1 | 1.2618E+1 |
| | Rank | **1** | 5 | 4 | 6 | 3 | 2 |
| F17 | Mean | 4.1152E+2 | **3.3937E+2** | 4.2809E+2 | 8.6913E+2 | 5.1843E+2 | 4.5848E+2 |
| | Std. | 1.1336E+2 | 1.3302E+2 | 1.9242E+2 | 1.8716E+2 | 3.3316E+1 | 3.2996E+1 |
| | Rank | 2 | **1** | 3 | 6 | 5 | 4 |
| F18 | Mean | **9.1202E+2** | 9.3195E+2 | 9.1219E+2 | 1.0997E+3 | 9.2694E+2 | 9.1744E+2 |
| | Std. | 2.3092E+0 | 1.9003E+1 | 3.3989E+0 | 1.0617E+2 | 1.8609E+0 | 1.0614E+0 |
| | Rank | **1** | 5 | 2 | 6 | 4 | 3 |
| F19 | Mean | **9.1346E+2** | 9.4374E+2 | 9.1639E+2 | 1.1074E+3 | 9.2699E+2 | 9.1739E+2 |
| | Std. | 4.4780E+0 | 3.5241E+1 | 1.9272E+1 | 1.0914E+2 | 2.2353E+0 | 1.1603E+0 |
| | Rank | **1** | 5 | 2 | 6 | 4 | 3 |
| F20 | Mean | **9.1137E+2** | 9.3208E+2 | 9.1231E+2 | 1.0822E+3 | 9.2738E+2 | 9.1708E+2 |
| | Std. | 1.6146E+0 | 2.7379E+1 | 2.0168E+0 | 7.7248E+1 | 2.1893E+0 | 1.2174E+0 |
| | Rank | **1** | 5 | 2 | 6 | 4 | 3 |
| F21 | Mean | **7.1840E+2** | 8.1505E+2 | 7.2512E+2 | 1.2791E+3 | 8.5482E+2 | 7.6159E+2 |
| | Std. | 2.4831E+2 | 1.9117E+2 | 2.7918E+2 | 1.2916E+2 | 3.5860E+1 | 2.8277E+1 |
| | Rank | **1** | 4 | 2 | 6 | 5 | 3 |
| F22 | Mean | **9.2496E+2** | 9.4015E+2 | 9.3961E+2 | 1.3126E+3 | 1.0474E+3 | 1.0161E+3 |
| | Std. | 3.8327E+1 | 4.8177E+1 | 4.0714E+1 | 2.1302E+2 | 1.6327E+1 | 1.4372E+1 |
| | Rank | **1** | 3 | 2 | 6 | 5 | 4 |

(*continued on next page*)

**Table 9** (*continued*).

| | | NS | PSO | M-PSO | CAPSO | LSHADE | jSO |
|---|---|---|---|---|---|---|---|
| F23 | Mean | **6.2683E+2** | 9.1955E+2 | 7.1282E+2 | 1.2446E+3 | 8.5767E+2 | 7.7977E+2 |
| | Std. | 1.7109E+2 | 2.1540E+2 | 2.5594E+2 | 1.0589E+2 | 4.2605E+1 | 2.8693E+1 |
| | Rank | **1** | 5 | 2 | 6 | 4 | 3 |
| F24 | Mean | 8.9105E+2 | 9.6696E+2 | 9.2988E+2 | 1.3029E+3 | 8.2673E+2 | **7.7922E+2** |
| | Std. | 2.4807E+2 | 7.4147E+1 | 2.2214E+2 | 2.2457E+2 | 4.1957E+1 | 4.8687E+1 |
| | Rank | 3 | 5 | 4 | 6 | 2 | **1** |
| F25 | Mean | 9.1076E+2 | 1.0755E+3 | 8.8001E+2 | 1.2403E+3 | 9.0035E+2 | **8.6318E+2** |
| | Std. | 2.3670E+2 | 9.8230E+1 | 2.3884E+2 | 1.5410E+2 | 5.3466E+1 | 5.3058E+1 |
| | Rank | 4 | 5 | 2 | 6 | 3 | **1** |

**Table 10**
Friedman mean rank for CEC2005 comparison from F1 to F25.

| | NS | PSO | M-PSO | CAPSO | LSHADE | jSO |
|---|---|---|---|---|---|---|
| 2 dimensions | 2.1 | 3.4 | 2.8 | 5.8 | 3.5 | 3.5 |
| 10 dimensions | 3.2 | 4.0 | 3.2 | 6.0 | 3.1 | 1.7 |
| 30 dimensions | 2.0 | 3.2 | 2.2 | 6.0 | 4.3 | 3.3 |

**Table 11**
Friedman mean rank for CEC2005 comparison for composition functions (F15 to F25).

| | NS | PSO | M-PSO | CAPSO | LSHADE | jSO |
|---|---|---|---|---|---|---|
| 2 dimensions | 1.3 | 3.9 | 3.0 | 6.0 | 3.5 | 3.5 |
| 10 dimensions | 3.0 | 4.6 | 4.0 | 6.0 | 2.3 | 1.1 |
| 30 dimensions | 1.8 | 4.1 | 2.7 | 6.0 | 3.8 | 2.5 |

**Table 12**
Algorithm complexity according to CEC2005 calculations [15].

| T0 | | T1 | | $\hat{T}2$ | $\frac{\hat{T}2-T1}{T0}$ |
|---|---|---|---|---|---|
| 8.5603 | 2 dimensions | 2.6736 | NS | 61.2966 | 6.8482 |
| | | | PSO | 16.5853 | 1.6251 |
| | | | M-PSO | 36.5573 | 3.9582 |
| | | | CAPSO | 39.3964 | 4.2899 |
| | | | LSHADE | 46.6621 | 5.1386 |
| | | | jSO | 42.6005 | 5.0085 |
| | 10 dimensions | 5.1178 | NS | 81.1183 | 8.8782 |
| | | | PSO | 37.6609 | 3.8016 |
| | | | M-PSO | 40.5768 | 4.1422 |
| | | | CAPSO | 42.1109 | 4.3215 |
| | | | LSHADE | 115.6112 | 12.9076 |
| | | | jSO | 118.6915 | 13.2674 |
| | 30 dimensions | 11.1488 | NS | 141.9046 | 15.2746 |
| | | | PSO | 43.9213 | 3.8284 |
| | | | M-PSO | 51.0528 | 4.6615 |
| | | | CAPSO | 48.5369 | 4.3676 |
| | | | LSHADE | 245.6049 | 27.3887 |
| | | | jSO | 251.5818 | 28.0869 |

**Table 13**
Inputs comparison between baseline and optimized case.

| | Number of holes [–] | Swirl number [–] | Injection pressure (bar) | EGR (%) |
|---|---|---|---|---|
| Baseline | 10 | 2 | 1800 | 0 |
| Optimized | 6 | 1.60 | 1898 | 14.57 |

noise emissions while keeping the $NO_x$ emission under control. The improved mixing conditions guaranteed by the optimized bowl correlates well with the shorter combustion duration and the improved performance, as shown previously.

The left-hand side of Fig. 14 shows the temperature contours of both cases. TDC snapshots show that in the optimized case the jet penetrates faster since the injection pressure is slightly higher and also because it has a bigger nozzle hole diameter. That promotes an improved jet-wall interaction with the piston surface distributing the flame in the combustion chamber. Moreover,

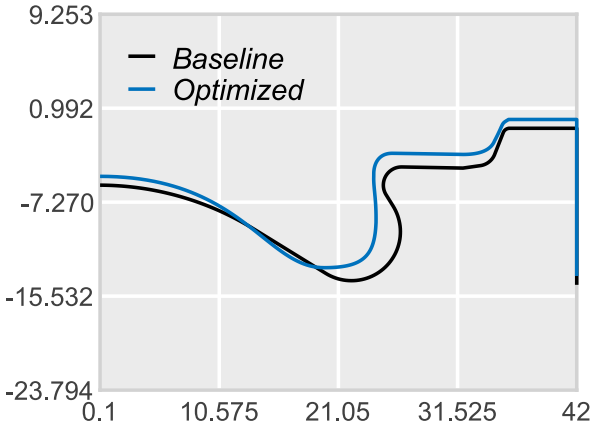



**Fig. 12.** Difference between baseline and optimized geometries.

**Table 14**
Output comparison between baseline and optimized case.

| | $NO_x$ [ppm] | Soot [ppm Mass] | ISFC [g/kWh] |
|---|---|---|---|
| Baseline | 1459.97 | 2.1e−01 | 191.07 |
| Optimized | 1443.35 | 1.9e−04 | 185.63 |

the optimized case presents a more homogeneous temperature distribution, thereby lowering the $NO_x$ production.

Finally, further studies are required to investigate the implementation to other operating conditions as maximum torque or partial loads in order to analyse if this optimized bowl profile would provide good results in terms of emissions and fuel consumption for those operating conditions. Also, subsequent optimization work is necessary to understand the influence of other input parameters as spray included angle, nozzle tip protrusion, start of injection, among others.

## 8. Conclusions

In this work, NS is proposed as an implementation of Novelty Search for the PSO algorithm. The CEC2005 benchmark is used to compare the behaviour of NS with PSO, M-PSO, CAPSO, LSHADE and jSO. The experiments have shown that NS has very good results in the optimum search at the expense of increasing the computational cost of the calculations even if few evaluations are required. Furthermore, NS stands out in composition functions, the more complex ones. Based on benchmark results, this algorithm is implemented in the optimization of a combustion system which has high dimensionality and demands few function evaluations. A methodology for coupling the Novelty Swarm and a computational fluid dynamic model of the combustion system is presented.

Several parameters that have a high impact on the combustion system were chosen for the optimization process. These parameters are geometrical variables that controls bowl shape, number of holes, swirl number, injection pressure and EGR rate. Computational tools were linked to obtain an automatic configuration

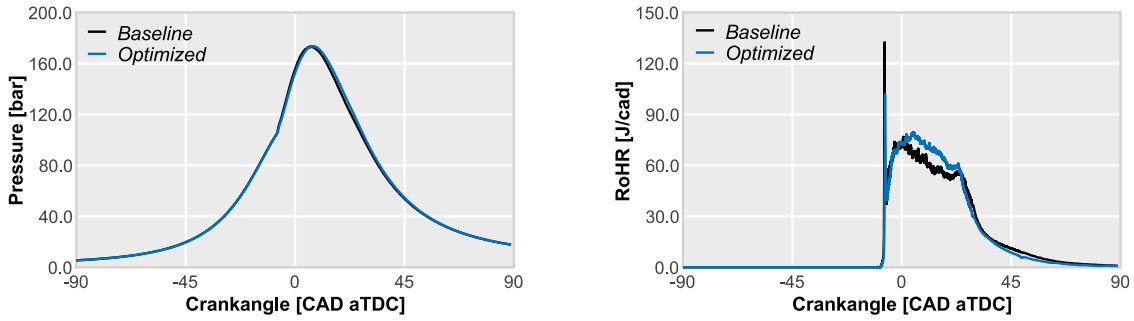



**Fig. 13.** Comparison between baseline results and optimized case. Left side: In-cylinder pressure evolution versus crank angle. Right side: Estimated Heat Release Rate versus crank angle.

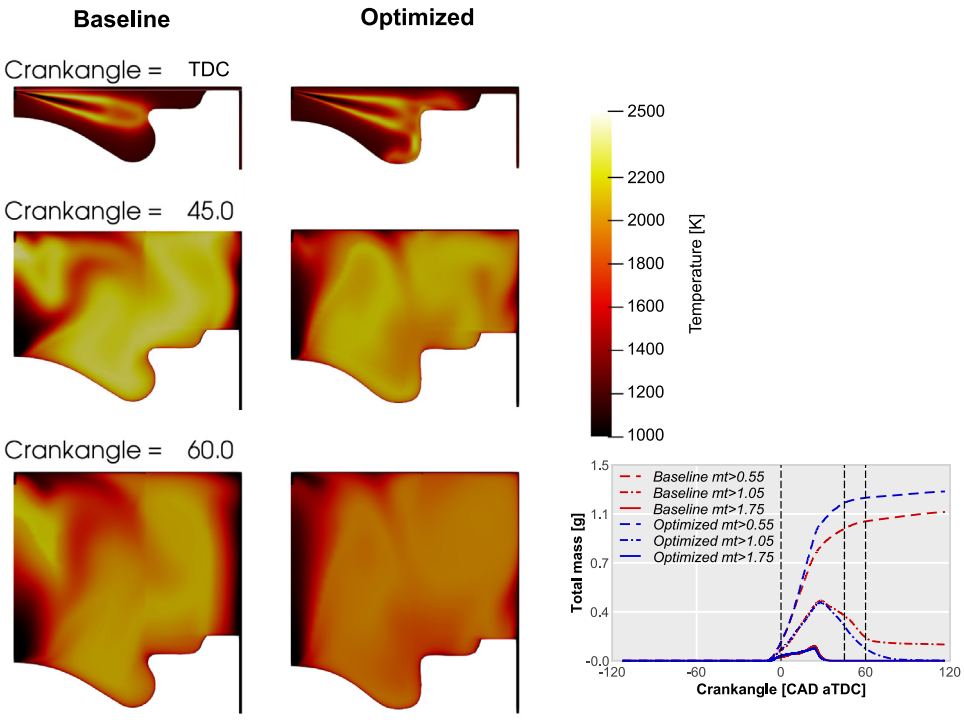



**Fig. 14.** Comparison between the baseline case and the optimum case. Left side: spatial distribution of temperature in the combustion chamber at three crank angles (Top Dead Centre, 45 and 60). Right side: evolution of the normalized mass with equivalence ratio over 0.55 (lean mixture), 1.05 (near to stoichiometric) and 1.75 (rich mixture). The vertical dashed lines correspond to the three instants depicted in the left side.

generator allowing to perform a thousand model evaluations automatically. The output targets of this work were pollutant emissions of $NO_x$ and soot and specific fuel consumption. These outputs were used to calculate the fitness function that guides the NS algorithm. The obtained results show that coupling CFD with NS can be employed as optimization method to the CI engines providing a new combustion system that reduces $NO_x$ and soot emissions and makes an improvement of 3% of fuel consumption.

Finally, this optimization method can be applied to different engine configurations and combustion concepts requiring just a few changes as the increment of the number of bowl geometrical parameters, the included spray angle or the use of a e-fuel for a full matching, which will be explored in future works.

## CRediT authorship contribution statement

**David Martínez-Rodríguez:** Validation, Data curation, Writing – original draft. **Ricardo Novella:** Formal analysis, Funding acquisition, Supervision. **Gabriela Bracho:** Investigation, Supervision, Writing – review & editing. **Josep Gomez-Soriano:** Formal analysis, Investigation. **Cassio Fernandes:** Validation, Data curation, Writing – original draft. **Tommaso Lucchini:** Software, Methodology. **Augusto Della Torre:** Software, Methodology. **Rafael-J. Villanueva:** Conceptualization, Methodology, Writing – review & editing. **J. Ignacio Hidalgo:** Conceptualization, Methodology, Funding acquisition.

## Declaration of competing interest

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

## Data availability

Data will be made available on request

## Acknowledgements

This work has been partially supported by

- This work has been supported by the grant PID2020-115270GB–I00 funded by MCIN/AEI/10.13039/501100011033;
- European Union FEDER funds;
- Spanish Ministerio de Economía y Competitividad through Grants PID2021-125549OB-I00 and PDC2022-133429-I00.
- The author C. S. Fernandes thanks the Universitat Politecnica de Valencia for his predoctoral contract (FPI-2019-S2-20-555), which is included within the framework of Programa de Apoyo para la Investigacion y Desarrollo (PAID).

## References

[1] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN 95 - International Conference on Neural Networks, IEEE, 1995.
[2] W. Liu, Z. Wang, X. Liu, N. Zeng, D. Bell, A novel particle swarm optimization approach for patient clustering from emergency departments, IEEE Trans. Evol. Comput. (2018).
[3] M.R. AlRashidi, M.E. El-Hawary, A survey of particle swarm optimization applications in electric power systems, IEEE Trans. Evol. Comput. 13 (4) (2009) 913–918.
[4] Hoai Bach Nguyen, Bing Xue, Peter Andreae, PSO with surrogate models for feature selection: static and dynamic clustering-based methods, Memet. Comput. 10 (3) (2018) 291–300.
[5] M.R. Bonyadi, Z. Michalewicz, Impacts of coefficients on movement patterns in the particle swarm optimization algorithm, IEEE Trans. Evol. Comput. 21 (3) (2017) 378–390.
[6] V. Kadirkamanathan, K. Selvarajah, P.J. Fleming, Stability analysis of the particle dynamics in particle swarm optimizer, IEEE Trans. Evol. Comput. 10 (3) (2006) 245–255.
[7] Pavel Novoa-Hernández, Carlos Cruz Corona, David A Pelta, Efficient multi-swarm PSO algorithms for dynamic environments, Memet. Comput. 3 (3) (2011) 163.
[8] Jin Gou, Yu-Xiang Lei, Wang-Ping Guo, Cheng Wang, Yi-Qiao Cai, Wei Luo, A novel improved particle swarm optimization algorithm based on individual difference evolution, Appl. Soft Comput. 57 (2017) 468–481.
[9] Joel Lehman, Kenneth O. Stanley, Exploiting open-endedness to solve problems through the search for novelty, Artif. Life - ALIFE (2008).
[10] Pablo Moscato, et al., On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms, in: Caltech Concurrent Computation Program, C3P Report, Vol. 826, 1989, p. 1989.
[11] Joel Lehman, Kenneth O. Stanley, Abandoning objectives: Evolution through the search for novelty alone, Evol. Comput. 19 (2011) 189–223.
[12] Joel Lehman, Kenneth O. Stanley, Efficiently evolving programs through the search for novelty, in: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, ACM Press, 2010.
[13] Joel Lehman, Kenneth O. Stanley, Novelty search and the problem with objectives, in: Genetic Programming Theory and Practice IX, Springer New York, 2011, pp. 37–56.
[14] Junfeng Chen, Ziwu Ren, Xinnan Fan, Particle swarm optimization with adaptive mutation and its application research in tuning of PID parameters, in: 2006 1st International Symposium on Systems and Control in Aerospace and Astronautics, IEEE, 2006.
[15] P.N. Suganthan, N. Hansen, J.J. Liang, K. Deb, Y.P. Chen, A. Auger, S. Tiwari, Problem definitions and evaluation criteria, in: CEC 2005 Special Session on Real Parameter Optimization, 2005, URL http://www.ntu.edu.sg/home/epnsugan/index_files/cec-05/cec05.htm.
[16] H. Wang, L.L. Zuo, J. Liu, W.J. Yi, B. Niu, Ensemble particle swarm optimization and differential evolution with alternative mutation method, Nat. Comput. (2018).
[17] Ruhul A. Sarker, Saber M. Elsayed, Tapabrata Ray, Differential evolution with dynamic parameters selection for optimization problems, IEEE Trans. Evol. Comput. 18 (5) (2014) 689–707.
[18] M.G. Epitropakis, V.P. Plagianakos, M.N. Vrahatis, Evolving cognitive and social experience in particle swarm optimization through differential evolution: A hybrid approach, Inform. Sci. 216 (2012) 50–92.
[19] Gautam Kalghatgi, Is it really the end of internal combustion engines and petroleum in transport? Appl. Energy 225 (2018) 965–974, URL http://www.sciencedirect.com/science/article/pii/S0306261918307955.
[20] Ka In Wong, Pak Kin Wong, Chun Shun Cheung, Chi Man Vong, Modelling of diesel engine performance using advanced machine learning methods under scarce and exponential data set, Appl. Soft Comput. 13 (11) (2013) 4428–4441, URL http://www.sciencedirect.com/science/article/pii/S1568494613001956.
[21] Timothy Johnson, Ameya Joshi, Review of Vehicle Engine Efficiency and Emissions, in: SAE Technical Paper Series, vol. 1, 2018, pp. 1–23, URL http://papers.sae.org/2017-01-0907/.
[22] Zhenhao Tang, Zijun Zhang, The multi-objective optimization of combustion system operations based on deep data-driven models, Energy 182 (2019) 37–47.
[23] Sudipto Chaki, Tapas Kumar Biswas, An ANN-entropy-FA model for prediction and optimization of biodiesel-based engine performance, Appl. Soft Comput. 133 (2023) 109929.
[24] Vijay Manikandan Janakiraman, Xuanlong Nguyen, Dennis Assanis, Nonlinear identification of a gasoline HCCI engine using neural networks coupled with principal component analysis, Appl. Soft Comput. 13 (5) (2013) 2375–2389.
[25] Kai Gaukel, Patrick Dworschak, Dominik Pélerin, Martin Härtl, Georg Wachtmeister, Combustion process optimization for oxymethylene ether fuels in a heavy–duty application, in: Johannes Liebl, Christian Beidl, Wolfgang Maus (Eds.), Internationaler Motorenkongress 2019, Springer Fachmedien Wiesbaden, Wiesbaden, 2019, pp. 351–367.
[26] Antonio J. Torregrosa, Alberto Broatch, Xandra Margot, Josep Gomez-Soriano, Understanding the unsteady pressure field inside combustion chambers of compression-ignited engines using a computational fluid dynamics approach, in: International Journal of Engine Research, SAGE Publications Ltd, 2018.
[27] Bambang Wahono, Ardhika Setiawan, Ocktaeck Lim, Experimental study and numerical simulation on in-cylinder flow of small motorcycle engine, Appl. Energy 255 (2019) 113863, URL https://www.sciencedirect.com/science/article/pii/S0306261919315508.
[28] Gilles Decan, Stijn Broekaert, Tommaso Lucchini, Gianluca D'Errico, Jan Vierendeels, Sebastian Verhelst, Evaluation of wall heat flux calculation methods for CFD simulations of an internal combustion engine under both motored and HCCI operation, Appl. Energy 232 (2018) 451–461, URL https://www.sciencedirect.com/science/article/pii/S0306261918315356.
[29] J. Mohammadhassani, A. Dadvand, Sh. Khalilarya, M. Solimanpur, Prediction and reduction of diesel engine emissions using a combined ANN–ACO method, Appl. Soft Comput. 34 (2015) 139–150, URL http://www.sciencedirect.com/science/article/pii/S1568494615002963.
[30] Alberto Broatch, Ricardo Novella, Josep Gomez-Soriano, Pinaki Pal, Sibendu Som, Numerical methodology for optimization of compression-ignited engines considering combustion noise control, 2018, http://dx.doi.org/10.4271/2018-01-0193.
[31] Aaron M. Bertram, Qiang Zhang, Song-Charng Kong, A novel particle swarm and genetic algorithm hybrid method for diesel engine performance optimization, Int. J. Engine Res. 17 (7) (2016) 732–747.
[32] Hadis Derikvand, Mohammad Shafiey Dehaj, Hadi Taghavifar, The effect of different sampling method integrated in NSGA II optimization on performance and emission of diesel/hydrogen dual-fuel CI engine, Appl. Soft Comput. 128 (2022) 109434.
[33] Pourya Rahnama, Majid Arab, Rolf D. Reitz, A Time-Saving Methodology for Optimizing a Compression Ignition Engine to Reduce Fuel Consumption through Machine Learning, SAE Int. J. Engines 13 (2) (2020) 267–288.
[34] Ahmed F. Ali, Mohamed A. Tawhid, A hybrid particle swarm optimization and genetic algorithm with population partitioning for large scale optimization problems, Ain Shams Eng. J. 8 (2) (2017) 191–206.
[35] Michael G Epitropakis, Vassilis P Plagianakos, Michael N Vrahatis, Evolving cognitive and social experience in particle swarm optimization through differential evolution: a hybrid approach, Inform. Sci. 216 (2012) 50–92.
[36] Wei-Neng Chen, Jun Zhang, Ying Lin, Ni Chen, Zhi-Hui Zhan, Henry Shu-Hung Chung, Yun Li, Yu-Hui Shi, Particle swarm optimization with an aging leader and challengers, IEEE Trans. Evol. Comput. 17 (2) (2013) 241–258.
[37] Chaoli Sun, Yaochu Jin, Ran Cheng, Jinliang Ding, Jianchao Zeng, Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems, IEEE Trans. Evol. Comput. 21 (4) (2017) 644–660.
[38] Zahra Beheshti, Siti Mariyam Hj Shamsuddin, CAPSO: centripetal accelerated particle swarm optimization, Inform. Sci. 258 (2014) 54–79.
[39] Amir Hossein Gandomi, Gun Jin Yun, Xin-She Yang, Siamak Talatahari, Chaos-enhanced accelerated particle swarm optimization, Commun. Nonlinear Sci. Numer. Simul. 18 (2) (2013) 327–340.
[40] Ryoji Tanabe, Alex S. Fukunaga, Improving the search performance of SHADE using linear population size reduction, in: 2014 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2014.
[41] Adam P. Piotrowski, L-SHADE optimization algorithms with population-wide inertia, Inform. Sci. 468 (2018) 117–141.
[42] Janez Brest, Mirjam Sepesy Maucec, Borko Boskovic, Single objective real-parameter optimization: Algorithm jSO, in: 2017 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2017.
[43] Antonios Liapis, Georgios N. Yannakakis, Julian Togelius, Constrained novelty search: A study on game content generation, Evol. Comput. 23 (1) (2015) 101–129.
[44] Iztok Fister, Andres Iglesias, Akemi Galvez, Javier Del Ser, Eneko Osaba, Iztok Fister, Matjaž Perc, Mitja Slavinec, Novelty search for global optimization, Appl. Math. Comput. 347 (2019) 865–881.

[45] Enrique Naredo, Leonardo Trujillo, Searching for novel clustering programs, in: Proceeding of the Fifteenth Annual Conference on Genetic and Evolutionary Computation Conference - GECCO '13, ACM Press, 2013.
[46] Marius Zubel, Tamara Ottenwälder, Benedikt Heuser, Stefan Pischinger, Combustion system optimization for dimethyl ether using a genetic algorithm, Int. J. Engine Res. 22 (1) (2019) 22–38.
[47] Jihad Badra, Fethi khaled, Jaeheon Sim, Yuanjiang Pei, Yoann Viollet, Pinaki Pal, Carsten Futterer, Mattia Brenner, Sibendu Som, Aamir Farooq, Junseok Chang, Combustion system optimization of a light-duty GCI engine using CFD and machine learning, in: SAE Technical Paper Series, SAE International, 2020.
[48] Gerardo Beni, Jing Wang, Swarm intelligence in cellular robotic systems, in: Paolo Dario, Giulio Sandini, Patrick Aebischer (Eds.), Robots and Biological Systems: Towards a New Bionics?, Springer Berlin Heidelberg, Berlin, Heidelberg, 1993, pp. 703–712.
[49] Jens Krause, Graeme D. Ruxton, Stefan Krause, Swarm intelligence in animals and humans, Trends Ecol. Evol. 25 (1) (2010) 28–34.
[50] Federico Marini, Beata Walczak, Particle swarm optimization (PSO). a tutorial, Chemometr. Intell. Lab. Syst. 149 (2015) 153–165.
[51] H.G. Weller, G. Tabor, H. Jasak, C. Fureby, A tensorial approach to computational continuum mechanics using object-oriented techniques, Comput. Phys. 12 (6) (1998) 620–631.
[52] G. Montenegro, A. Onorati, F. Piscaglia, G. D'Errico, Integrated 1D-MultiD fluid dynamic models for the simulation of ICE Intake and exhaust systems, in: SAE Technical Paper Series, vol. 1, 2007, pp. 776–790.
[53] T. Lucchini, G. D'Errico, H. Jasak, Z. Tukovic, Automatic mesh motion with topological changes for engine simulation, in: SAE Technical Paper Series, vol. 1, 2007, pp. 1–20.
[54] H.M. Ismail, H.K. Ng, S. Gan, T. Lucchini, Approach for the Modeling of Reacting Biodiesel Fuel Spray using OpenFOAM, in: SAE Technical Paper Series, vol. 1, 2014, pp. 1–9.
[55] G. D'Errico, T. Lucchini, G. Hardy, F. Tap, G. Ramaekers, Combustion Modeling in Heavy Duty Diesel Engines Using Detailed Chemistry and Turbulence-Chemistry Interaction, in: SAE Technical Paper Series, vol. 1 (2015) 1–14.
[56] Raul Payri, Jaime Gimeno, Ricardo Novella, Gabriela Bracho, On the rate of injection modeling applied to direct injection compression ignition engines, Int. J. Engine Res. 17 (10) (2016) 1015–1030.
[57] Harun Mohamed Ismail, Hoon Kiat Ng, Suyin Gan, Tommaso Lucchini, Angelo Onorati, Development of a reduced biodiesel combustion kinetics mechanism for CFD modelling of a light-duty diesel engine, Fuel 106 (2013) 388–400.
[58] Harun Mohamed Ismail, Hoon Kiat Ng, Xinwei Cheng, Suyin Gan, Tommaso Lucchini, Gianluca D'Errico, Development of thermophysical and transport properties for the cfd simulations of in-cylinder biodiesel spray combustion, Energy & Fuels 26 (8) (2012) 4857–4870.
[59] Victor Yakhot, Steven A. Orszag, Renormalization-group analysis of turbulence. 57(14):1722–1724, 1986.
[60] C. Angelberger, T. Poinsot, B. Delhay, Improving near-wall combustion and wall heat transfer modeling in SI engine computations, in: SAE Technical Paper Series, vol. 1, 1997.
[61] G. D'Errico, T. Lucchini, F. Contino, M. Jangi, X.-S. Bai, Comparison of well-mixed and multiple representative interactive flamelet approaches for diesel spray combustion modelling, Combust. Theory Model. 18 (1) (2014) 65–88.
[62] Jesus Benajes, Ricardo Novella, Jose Manuel Pastor, Alberto Hernández-López, Manabu Hasegawa, Naohide Tsuji, Masahiko Emi, Isshoh Uehara, Jordi Martorell, Marcos Alonso, Optimization of the combustion system of a medium duty direct injection diesel engine by combining CFD modeling with experimental validation, Energy Convers. Manage. 110 (2016) 212–229.
[63] Tommaso Lucchini, Augusto Della Torre, Gianluca D'Errico, Gianluca Montenegro, Marco Fiocco, Amin Maghbouli, Automatic Mesh Generation for CFD Simulations of Direct-Injection Engines, in: SAE Technical Paper Series, vol. 1, 2015.
[64] Raul Payri, Gabriela Bracho, Jaime Gimeno, Abian Bautista, Rate of injection modelling for gasoline direct injectors, Energy Convers. Manage. 166 (2018) 424–432.
[65] J. Benajes, P. Olmeda, J. Martín, R. Carreño, A new methodology for uncertainties characterization in combustion diagnosis and thermodynamic modelling, Appl. Therm. Eng. 71 (2014) 389–399.
[66] Python programming language - , 2020, URL https://www.python.org, [Online; accessed 31 January 2020].
[67] Package for scientific computing with Python - , 2020, URL https://numpy.org, [Online; accessed 31 January 2020].
[68] James Kennedy, Russell C. Eberhart, Yuhui Shi, Swarm Intelligence, Elsevier, 2001.
[69] Ender Ozcan, Chilukuri K. Mohan, Analysis of a simple particle swarm optimization system, Intell. Eng. Syst. Through Artif. Neural Netw. 8 (1998) 253–258.
[70] Joaquín Derrac, Salvador García, Daniel Molina, Francisco Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (1) (2011) 3–18.