An asynchronous steady-state NSGA-II algorithm for multi-objective optimization of Diesel combustion

Mouadh Yagoubi\textsuperscript{1,2}, Ludovic Thobois\textsuperscript{1}, Marc Schoenauer\textsuperscript{2}

\textsuperscript{1} PSA PEUGEOT CITROEN, Centre technique de Vlizy, Route de Gisy 78943 Vlizy-Villacoublay, France.
\textsuperscript{2} Projet TAO, INRIA Saclay & LRI, Universit\'e Paris Sud, Orsay, France

Abstract

In order to comply with environmental regulations, automotive manufacturers have to develop efficient engines with low fuel consumption and low emissions. Thus, development of engine combustion systems (chamber, injector, air loop) becomes a hard task since many parameters have to be defined in order to optimize many objectives in conflict. Evolutionary Multi-objective optimization represents an efficient tool to explore the search space and find promising engine combustion systems. In this work, a phenomenological 0D model of Diesel engine combustion is coupled with a state-of-the-art multi-objective evolutionary algorithm NSGA-II. The 0D combustion model allows to compute, from the compression to the expansion strokes, the phenomena of: Injection, mixing, combustion, and major pollutants formation, thanks to a detailed kinetic scheme. Besides, it takes into account spatial heterogeneities on temperature and mixture composition inside the combustion chamber. A standard parallel version of NSGA-II based on master-slave paradigm is applied using a Grid system. In this standard version, the selection process is generational: Variation operators are applied once all individuals belonging to the same generation have been evaluated. However, the computation of different solutions can have very different durations. Thus, some processors sometimes stay idle. To tackle this issue, a steady-state version of NSGA-II is implemented: Only one individual is created per generation, and uninterrupted use of available processors in the Grid system is ensured. Comparison between both approaches is performed.

Keywords: Multi-objective optimization, Asynchronous steady-state NSGA-II, Diesel combustion.

\section{1 Introduction}

Due to environmental regulations, the development of efficient automotive engines with low fuel consumption and low pollutant emissions becomes a hard task for engine designers. In order to explore more advanced engine technologies and more engine designs, numerical computation plays a more and more important role in the development of engine processes but several limitations remain. On the one hand, the complexity of phenomena that occurs in the combustion chamber involves a great number of devices in the numerical model which increases considerably the simulation duration. On the other hand, objectives that are to be optimized are naturally in conflict. Multi-objective optimization \cite{6} (optimization of more than one objective simultaneously) using Genetic Algorithms has been widely used in the field of engine research. However, it requires a lot of evaluations to converge toward the Pareto front. Thus, the computation cost of the simulation approach increases considerably. The most well-known algorithms in the field of multi-objective genetic algorithms are: NSGA-II \cite{6}, SPEA-II \cite{15}, IBEA \cite{14}, and EpsilonMOEA \cite{5}.

Evolutionary Algorithms are “generate and test” stochastic optimization algorithms that address the usual trade-off between exploitation of best-so-far solutions and exploration of yet unexplored regions of the search space by relying on a crude version of the Darwinina paradigm of survival of the fittest. At each iteration of the algorithm, selection operators are biased toward the best-performing individuals, whereas the variation operators (crossover and mutation) perform random moves in the search space. For single-objective optimization, the quality of the individuals is given by the objective value. For the multi-objective case, more sophisticated fitness has to be designed (see Section 4).

The behavior of Evolutionary Algorithms is controlled by several parameters. One of those parameters is the number of offspring that are generated at each generation, ranging from 1 to several times the parent population size, and another parameter is the selection procedure that is used to select the new population from the old parents and newborn offspring. Historically, Genetic Algorithms use the so-called
generational model, where the number of offspring is the number of parents, and all offspring replace all parents. Another popular method, called Steady-state GA, creates one single offspring, and inserts it in the parent population after removing one poorly performing one [11].

In this paper, we focus on the master/slave parallelization [2] of NSGA-II algorithm coupled to a pseudo 0D model using detailed kinetic scheme. If this model allows a good compromise between CPU cost and Diesel combustion modeling details, its return time is significant and different solutions can have very different durations. In such context, using a generational scheme requires some synchronization of all fitness computations before the new generation can be designated, and some processors will stay idle. In order to reduce the global duration of the optimization and to ensure an efficient use of all processors, a parallel steady-state asynchronous algorithm is designed, based on NSGA-II, one of the most popular EMOA: when a node has completed the evaluation of an individual, a new individual is immediately sent for evaluation.

The remainder of this paper is structured as follows. Section 2 presents some works using Genetic Algorithms in the field of engine design research. Section 3 briefly describes the 0D Diesel combustion model used here. Section 4 describes the standard NSGA-II algorithm and its asynchronous steady-state version. The optimization loop is presented in Section 5. Section 6 discusses the obtained results, while Section 7 summarizes the contribution of this work and suggests directions for future research.

2 Related work

In this section we presents some previous works which have made use of genetic algorithms for the optimization of 3D engine simulation models.

Senecal et al. proposed in [12] a methodology for internal combustion engine design which incorporates multi-dimensional modelling and experiments to optimize Diesel engine combustion and emissions formation. The full-cycle engine simulation code was coupled to a Genetic Algorithm. The study simultaneously investigated the effects of six engine input parameters on emissions and performance for a high-speed medium-load operating point.

Hiroyasu et al. presented in [8] an application of multi-objective optimization to a 0D emission problem of a Diesel engine. The objectives to be minimized were: NOx, Soot, and Specific fuel consumption. However, the minimization of these objectives was done by varying only one design variable which is the injection rate. Hiroyasu et al. proposed and extension of previous work in [7], taking into account several design variables such as: Air swirl, EGR, injection timing and multiple injections.

De Risi et al. investigated in [3] the effect of combustion chamber shape and injection strategy on Diesel engine emissions (NOx, soot, and HC) without significantly decreasing engine performance. This was achieved by using an optimization process based on a genetic algorithm and a 0D model. A penalty function was introduced to control engine performance.

Summarizing this section, many of the works in the literature were interested in using multi-objective genetic algorithms for reducing emissions and fuel consumption. However, all algorithms used in the literature have a generational model, which increases the cost of the global optimization when the individuals of the same population have a very different evaluation duration. Besides, optimization using 3D models are very expensive, and 0D models for which the optimization cost is reasonable do not take into account some phenomena like mixture modeling.

3 Combustion model description

PDF0D is an internal PSA tool for the simulation of the Diesel combustion at part load operating points. It allows the computation of the thermochemical evolution of gas mixture during a complete engine cycle. The mixture is described by numerous particles with a specific temperature and composition. The PDF0D tool computes the evolution of temperature and composition of all particles taking into account compression and expansion strokes, injection, mixing between particles, combustion, and wall heat loss. In this study one hundred particles are used to model the spatial heterogeneities inside the combustion chamber. Initial mixing is imposed by a joint Probability Density Function (PDF), a Gaussian function for the composition given by the equivalence ratio(Φ), the Exhaust gas recirculation ratio (EGR), and the temperature. The equivalence ratio represents the ratio of fuel and air compared to the stoechiometric ratio which allows a complete combustion of the fuel thanks to the exact quantity of air needed.

The injection is divided into several sub-injections. For each sub-injection, a pure fuel particle is added
with a mass calculated thanks to the injection rate. Figure 1 shows the different steps of the PDF0D model from the input (initial conditions parameters) to the output (pollutants formation).

Figure 1: Diagram of the Diesel combustion model

**Thermodynamic computation**  At each time step, thermodynamic variables such mean temperature and pressure inside the cylinder are computed using the law of perfect gas (equation (1)) for each particle and their internal energy equation (equation (2)):

\[
P V = nRT
\]

\[
e = h - P/\rho = \int_{T_0}^{T} C_v dT - \frac{RT_0}{W} + \sum_{k=1}^{n} \Delta h_{f,k} Y_k
\]

**Mixture**  This step simulates the effect of the turbulence on the mixing between particles, which is modelled by the modified Curl’s mixing model [9]. Pairs of particles (denoted by \( p \) and \( q \)) are randomly selected and their compositions are computed as follows:

\[
\phi_{P,new}^{P} = \phi^{P} + \frac{1}{2} a (\phi^{Q} - \phi^{P})
\]

\[
\phi_{Q,new}^{Q} = \phi^{Q} + \frac{1}{2} a (\phi^{P} - \phi^{Q})
\]

The mixing time constant is directly linked to turbulent time scale, a user parameter. In this study, this parameter is decomposed during the cycle so as to distinguish three turbulence sources: internal aerodynamics linked to chamber design (ENGINE-FREQ), turbulence due to injection (FACT-FREQ) and turbulence heterogeneities between the highest turbulent zones near the spray and the zone far away (FACT-INJ).
**Heat lost** PDF0D takes into account the gas-to-wall heat transfer process. A number of correlations have been proposed for calculating the instantaneous heat transfer coefficient. PDF0D uses the Woschni correlation model [13] which has frequently been used for engine studies.

**Injection** The user chooses in the initial configuration the crank-angle corresponding to the injection. At this crank-angle, a new Gaussian distribution of the particles is imposed by PDF0D which takes into account the injected mass.

**Combustion** Combustion is computed with Hewson kinetic scheme [1] and the Chemkin-II package [10]. The kinetic scheme describes in details the oxidation of n-heptane (modelling real Diesel) with oxygen. All the unburnt hydrocarbon species (HC) during this process is available, as well as carbon oxides (CO and CO$_2$). A nitrogen oxides scheme has been added to the hewson’s scheme in order to access to nitrogen oxide (NO). Thanks to these schemes and the chemkin-II library, at each time step, the mass reaction rates for all species and for all particles are calculated. For species $k$, this rate $\omega_k$ is the sum of rates $\omega_{kj}$ produced by all $M$ reactions:

$$\omega_k = \Sigma_{j=1}^{M} \omega_{kj}$$  \hspace{1cm} (5)

4 **NSGA-II algorithms**

This Section describes the base NSGA-II algorithm [6] and its asynchronous steady-state variant. NSGA-II is the most popular Evolutionary Multiobjective Optimization Algorithm (MOEA). It evolves a population of possible solutions according to the basic principles of (single-objective) Evolutionary Algorithms (EAs), but uses a multi-objective ranking of the individuals to perform selection, instead of the usual ranking based on (single) fitness value. Let us first introduce rapidly EAs, then give some definitions about multi-objective ranking.

An Evolutionary Algorithm evolves a population of fixed size $N$ using discrete time steps, also called generations. Let $P(t)$ denote the population at time $t$. The first population $P_0$ is created randomly, generally uniformly on the search space. At time $t$, a given number of offspring $O$ is created from the parent population $P_t$: Each offspring is created by first selecting 2 parents from $P_t$ using parental selection, applying crossover with probability $p_c$ to these parents (leaving parents unchanged with probability $1-p_c$), uniformly selecting one of the children of this crossover and applying mutation with probability $p_m$ to generate the final offspring. Population $P_{t+1}$ is then selected applying survival selection to either the $O$ offspring or to the $N + O$ parents + offspring.

The most popular parental selection, that is used in NSGA-II, is tournament selection: given a size $T$ of the tournament, tournament selection first uniformly choses $T$ individuals from the population, and returns the best one according to the fitness values. Large $T$ increase the chances of selecting highly fit individuals, while small $T$ give a chance to poorly performing individuals to be nevertheless selected. Survival selection is very often deterministic, the survivors being the $N$ best individuals.

An EMOA can be obtained from the same general template provided a comparison procedure can be designed to take into account the multiple objectives. In order to do so, let us recall some definitions:

**Definition 1** [4]: Concept of dominance
A solution $x^{(1)}$ is said to dominate another solution $x^{(2)}$, if both following conditions are true:
1. The solution $x^{(1)}$ is not worse than $x^{(2)}$ for all objectives.
2. The solution $x^{(1)}$ is strictly better than $x^{(2)}$ for at least one objective.

**Definition 2** [4]: Pareto Set
Among a set of solutions $P$, the Pareto (or non-dominated) set of solutions $P'$ are those that are not dominated by any member of the set $P$. The goal of multi-objective optimization is to identify the Pareto set of the problem at hand.

However, the dominance relation only induces a partial order on the solutions, and a secondary criterion is needed in order to be able to use tournament or deterministic selection. This secondary criterion is generally the degree of innovation of the individual, i.e. whether or not this individual is important
for the diversity of the population. In NSGA-II, this diversity secondary criterion is implemented using the **crowding distance**: for each individual, its crowding distance is the volume of the maximal hypercube in objective space that does not contain any other point of the population. Individuals that are isolated are hence favored against individuals that lie in crowded regions of the objective space. Comparing two individuals thus amounts to first looking if one dominates the other. If yes, the dominant one is returned as best. If not, the one with largest crowding distance is returned as best.

**Synchronous Generational NSGA-II** In the original NSGA-II [6], exactly $N$ offspring are generated at each generation. The survival selection involves the $2N$ parents plus offspring, out of which the best $N$ survive (according to the multiobjective comparison described above). When a master-slave parallel version is run and the fitness computations are distributed among the slaves, this algorithm requires synchronisation at every generation, like standard single-objective parallel EAs [2], and fast processors, or processors haveing received individuals that are fast to evaluate will have to wait for the slowest ones.

**Asynchronous Steady-state version of the NSGA-II algorithm** A steady state version of NSGA-II uses $O = 1$, i.e., generates a single individual at each “generation”. The survival selection amounts to remove the worst parent, and insert the newborn offspring instead. When running in parallel, the asynchronous version inserts whichever individual comes back first, and immediately sends a newborn offspring for evaluation [2]. Hence no processor has to wait. However, the dynamics of the parallel version of the algorithm is different from that of the sequential one, biasing the comparison of their respective performances.

## 5 Optimization loop

We describe in this section the coupling made between the NSGA-II algorithm and the PDF0D model. An application of the master-slave paradigm to NSGA-II lies in evaluating the offspring population in parallel. If the population size is $N$, the same number of processors is needed on the grid system. The PDF0D code communicates with the NSGA-II algorithm through an input-output file system. Initially, the master executes the code of the algorithm. For each individual, the algorithm generates the input file and evaluates the objective function on a processor in the grid system. When the evaluation is completed, the PDF0D code returns the output file with objective function values.

For the PDF0D configuration, we choose a medium-load operating point with an engine speed = 2400 rpm and 13.66. We use 10 input parameters for the optimization process. Table 1 describes the meaning of each parameter and its corresponding upper and lower bounds. These parameters can be classified into three categories: Air loop parameters (EGR, P2, and T2), mixture parameters (ENGINE-FREQ, FACT-FREQ, FACT-INJ, and RATE-MAX) , and injection parameters (SOI, PERMEA, PRAIL).

Three objectives are to be minimized, namely : NOx, HCCO, and specific fuel consumption. In order that the individuals stemming from the optimization process be comparable in terms of objectives values, they must have the same power produced by the engine cycle. Since all individuals have the same RPM, having the same power means that the individuals must have the same Indicated Mean Effective Pressure (IMEP). The target value of the IMEP is defied according to experiments on a real engine. A Gradient descent algorithm is coupled to PDF0D to converge toward the target value of the IMEP. For each PDF0D evaluation, a maximum of 4 sub-iterations of gradient algorithm is performed. At each sub-iteration, the injected mass is adapted by the gradient algorithm to converge toward the IMEP target value. IMEP is then treated as a constraint in the genetic algorithm, and individuals that fail to converge toward the IMEP target value, are penalized. The penalization depends on the difference between the obtained value and the target value of the IMEP.

In this study, we compare two optimization methods: the first one uses a standard parallel form of NSGA-II with generational model, and the second one use an asynchronous steady-state model of NSGA-II. Figure 2 and figure 3 describe the generational and the asynchronous steady-state mechanisms respectively.

For both approaches, we use a population of size 40 (40 is the number of available processors on our grid system), a crossover probability of 0.8, a mutation probability of $1/n$ (where $n$ is the number of variables). The parameters are treated as real variables and the simulated binary crossover(SBX) and
### Table 1: Input optimization parameters

<table>
<thead>
<tr>
<th>Input Parameter</th>
<th>Units</th>
<th>Description</th>
<th>Lower bound</th>
<th>Upper bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>EGR</td>
<td>%</td>
<td>Exhaust gas recirculation rate</td>
<td>0</td>
<td>0.6</td>
</tr>
<tr>
<td>P2</td>
<td>Bar</td>
<td>Intake pressure</td>
<td>1.6</td>
<td>3</td>
</tr>
<tr>
<td>T2</td>
<td>K</td>
<td>Intake temperature</td>
<td>380</td>
<td>480</td>
</tr>
<tr>
<td>SOI</td>
<td>crank angle</td>
<td>Start Of Injection</td>
<td>300</td>
<td>380</td>
</tr>
<tr>
<td>ENGINE-FREQ</td>
<td>Hz</td>
<td>Frequency of mixture before injection (turbulence created by the engine)</td>
<td>500</td>
<td>50000</td>
</tr>
<tr>
<td>FACT-FREQ</td>
<td>Hz</td>
<td>Frequency of mixture after injection (turbulence created by the injection)</td>
<td>ENGINE-FREQ</td>
<td>150000</td>
</tr>
<tr>
<td>FACT-INJ</td>
<td>/</td>
<td>Ratio between spray zone and the rest of the chamber</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>PERMEA</td>
<td>cm³/s/30s/1000bar</td>
<td>Thermodynamic flow of injection</td>
<td>300</td>
<td>600</td>
</tr>
<tr>
<td>PRAIL</td>
<td>Bar</td>
<td>Injection pressure</td>
<td>500</td>
<td>3000</td>
</tr>
<tr>
<td>RATE-MAX</td>
<td>%</td>
<td>the maximum ratio of mass that can be exchanged between 2 particles</td>
<td>$10^{-3}$</td>
<td>$10^{-1}$</td>
</tr>
</tbody>
</table>

the the real-parameter mutation operator are used. Both algorithms are run for 1600 evaluations (i.e. 40 generations in the generational variant).

### 6 Results

This section is devoted to evaluating the results of the optimizations. We first compare the global cost of the optimizations. Then we pay attention to the quality of the results by comparing both Pareto fronts. Finally, we analyse the parameters configurations of some individuals of the Pareto front.

#### 6.1 Global optimization cost

Table 2 shows the duration of the optimization at different numbers of function evaluation for both Standard and Asynchronous steady-state NSGA-II algorithms. The total optimization cost with a standard NSGA-II is 17 days and one hour, while that associated with the asynchronous steady-state is 9 days and 20 hours. The first (expected) result is that the asynchronous steady-state algorithm performs significantly more evaluations than the standard one for the same duration.

Figure 4 displays the different evaluation durations of individuals of the the first random generation. This confirms that individuals have a variable evaluations durations. We explain this by the fact that some PDF0D input parameters have a direct impact on the restitution time of the evaluation. On the one hand, a very large value of the frequency after injection parameter (FACT-FREQ) significantly slows down the execution of a time step, and consequently, increases the evaluation time. On the other hand, some individuals fail to have a combustion (Thermodynamics conditions in terms of Pressure and Temperature not met) such as individuals with a very late start injection (SOI after 370 crank-angle), so they return very quickly.

With the generational NSGA-II, the time needed to evaluate the population is the time that needed by the slowest individual. Note that individuals with very large evaluation time (that exceeds 8 hours) are killed and penalized to avoid extremely costly evaluations. In the asynchronous steady-state, no individual needs to be killed because optimization proceeds anyway. Note that the evaluation time of the slowest individual is 24 hours.
Table 2: comparison of optimization duration at different evaluations numbers

<table>
<thead>
<tr>
<th>Number of evaluations</th>
<th>Standard parallel NSGA-II</th>
<th>Asynchronous Steady-state NSGA-II</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>39.5</td>
<td>26.52</td>
</tr>
<tr>
<td>400</td>
<td>81.23</td>
<td>52.69</td>
</tr>
<tr>
<td>600</td>
<td>130.35</td>
<td>82.54</td>
</tr>
<tr>
<td>800</td>
<td>182.6</td>
<td>110.98</td>
</tr>
<tr>
<td>1000</td>
<td>253.32</td>
<td>138.13</td>
</tr>
<tr>
<td>1200</td>
<td>304.16</td>
<td>164.2</td>
</tr>
<tr>
<td>1400</td>
<td>351.1</td>
<td>200.2</td>
</tr>
<tr>
<td>1600</td>
<td>410.3</td>
<td>236.2</td>
</tr>
</tbody>
</table>

6.2 Comparing Pareto fronts

Figure 5 and Figure 6 show 2D Pareto fronts in the plane "specific fuel consumption-HCCO" times "specific fuel consumption-NOx". Both optimizations are comparable in terms of quality of convergence.
However, some interesting points in terms of NOx values (with worse values in HCCO) are present in the asynchronous steady-state optimization and not in the generational one. These points correspond to individuals with a very large value of the FACT-FREQ parameter. As explained above, individuals with large values of FACT-FREQ are killed in the generational approach because of their huge evaluation time. Thus, they do not appear in the results of the generational optimization although they are very interesting in terms of objectives values.

6.3 Parameters evolution

Figures 7 to 10 display the evolution of some parameters (SOI, P2, T2, EGR) during the optimization process with the asynchronous steady-state NSGA-II algorithm. For the air loop parameters, P2 goes to its upper value while T2 goes to its lower value. This can be explained as follows: On the one hand, the higher P2, the more mechanical work is recovered outside the combustion chamber, and consequently, the specific fuel consumption is lower. On the other hand, low T2 temperature decreases the maximum temperature of the cycle, which leads to a reduction of NOx.

The EGR evolution consists of two zones, one for large values of EGR rate (0.2-0.3), and the other for low value of EGR rate (0-0.1). We selected two arbitrary individuals of the Pareto front with different values
Figure 7: SOI

Figure 8: P2

Figure 9: T2

Figure 10: EGR

Figure 11: Pressure

Figure 12: Temperature
of EGR (0 and 0.26 respectively), and plotted their corresponding curves of Pressure, temperature, NO, and HCCO, as can be seen on Figures 11, 12, 13, and 14.

Combustion without EGR reaches very high temperature compared to combustion with EGR ratio of 0.26: EGR plays a role of thermal buffer and slows the combustion process.

For the injection parameters, the evolution of SOI is quickly bounded by 340 and 360: Late combustions generally fail to converge toward the target value of the IMEP, and have a high fuel consumption level, while early combustions are not so good in NOx because of high combustion temperature.

Mixture parameters does not have a direct impact on the optimization process, except FACT-FREQ parameter which goes to its upper value promoting individuals with good values in NOx.

7 Conclusions and Future Work

In this paper, we have presented an application of an asynchronous steady-state genetic algorithm to a combustion model (PDF0D). We have chosen a state-of-the-art multi-objective Genetic Algorithm: NSGA-II. Both the standard and an asynchronous steady-state version have been tested on the optimization of PDF0D.

The results have shown that the Pareto fronts are comparable in term of quality of solutions, but the global optimization cost is lower with an asynchronous steady-state scheme than with the standard version. The asynchronous steady-state version ensure the uninterrupted use of the processors on the grid when the individuals have very different evaluation durations.

For future work, and with the same goal of reducing the global cost of the optimization, we will consider using a surrogate model to reduce the total number of evaluations.

References


