A two-step optimization for crankshaft counterweights

Engineering Science and Technology, an International Journal 51 (2024) 101657

Contents lists available at ScienceDirect

Engineering Science and Technology, an International Journal

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

A two-step optimization for crankshaft counterweights

Eugenio Brusa, Alberto Dagna *, Cristiana Delprete, Chiara Gastaldi

Department of Mechanical and Aerospace Engineering, Politecnico di Torino, Corso Duca degli Abruzzi, 24, Torino, 10129, Italy

ARTICLE INFO

Keywords:
Crankshaft
Balancing
Geometry optimization
Multi-body

ABSTRACT

The optimization of industrial products and processes has been, since the beginning of the third industrial revolution, a fundamental aspect of the design phase as it allows, together with the testing and validation phases, the improvement of the performance or efficiency of what has been designed. Optimization is also applied to the counterweights of the crankshafts that ensure the balancing of the forces and moments generated by internal combustion engines during their regular use. Traditionally, this process is carried out, at the preliminary stage, in an analytical way, using specific formulas for each engine configuration. This approach, however, allows the identification of only the macro-parameters of the counterweight, i.e. mass and position of the center of gravity, leaving the designer the translation into technical drawing of the result of the optimization. Thanks to the increase in computational power obtained in recent decades and the interconnection of systems, typical of Industry 4.0, this work intends to propose a new methodology for optimizing counterweights, based on a two step approach, able to identify the best solution not only in terms of macro-parameters, but also of the specific dimensional parameters of the counterweight.

1. Introduction

Today's industrial reality is constantly looking for products and solutions that allow us to better meet customer demands. This is equally true for the automotive industry, which seeks to refine even the smallest detail to improve the performance and quality of its products [1]. One of these aspects is certainly the balancing of the crankshafts of internal combustion engines, a highly complex process that is difficult to carry out in the preliminary stages of development [2,3]. However, making a crankshaft as balanced as possible is extremely important: in fact, a good balance allows to greatly reduce the vibrations generated by the engine itself, improving passenger comfort [4]; in addition, it is possible to reduce the deformations induced on the shaft, ensuring a reduction in friction between moving components [5–7] and, consequently, an increase in the efficiency of the powertrain and a reduction in surface wear [8–10]. For this reason, it is imperative to identify new development methodologies, which allow to advance the design of balancing counterweights, the most used component to balance the crankshafts [11,12], to a more preliminary phase of engine development. This would pave the way for a faster and more flexible development process, as development time and costs tend to increase exponentially as the [13] project progresses. In fact, balancing procedures tend to be a very late-design phase, employing experimental procedures and on-product testing to find the correct balancing configuration. For instance, the influence coefficient balancing method [14] employs trial masses to identify an influence matrix, based on experimental measurements obtained through the in-place balancing procedure [15], from which a set of correction masses is identified. The same could be said for the modal balancing procedure [16], which analyses one mode at a time of the rotor system to find trial balancing masses to then test them in an experimental environment. Even a more complex approach, like the application of the least-square method for balancing optimization [17] is based on experimental testing. Meanwhile, more numerical approaches usually consider balancing masses just as point-masses, and rarely are developed for a wide variety of engine architecture [18–20]. The aim of this contribution is therefore to propose a new method for the development of counterweights. It is based on numerical script for the computation of the forces and moments generated by a crankshaft and a simplification of the geometry of the balancing counterweights, which is expressed as a function of a set of independent design parameters. This characterization enables the execution of an optimization process that evaluates the best solution in terms of mass, size and balancing. Optimization techniques are commonly employed in the industrial field and allow to find the best solution to a certain problem [21–24]. This is very in line with what the current contribution aims at achieving, since obtaining an optimized counterweight geometry allows to maximize its balancing capability and limiting at the same time the additional

* Corresponding author.
E-mail addresses: eugenio.brusa@polito.it (E. Brusa), alberto.dagna@polito.it (A. Dagna), cristiana.delprete@polito.it (C. Delprete), chiara.gastaldi@polito.it (C. Gastaldi).

https://doi.org/10.1016/j.jestch.2024.101657
Received 3 August 2023; Received in revised form 13 February 2024; Accepted 19 February 2024
Available online 23 February 2024

masses that must be applied to the crankshaft. Such process will be done through a preliminary integration of simulation tools, which will provide a faster and more intuitive development infrastructure for the designer, which includes both the geometric characterization of the counterweight and the definition of its configuration on the crankshaft, and the optimization of its parameters. The tools that will be used for this purpose are the following:

- MATLAB®, for calculating the geometry of the counterweight and its forces and balancing moments;
- DAKOTA, specialized in the implementation of optimization algorithms [25].

Due to the multiple phases of an engine development, it is quite improbable that a counterweight design proposed at the preliminary stage, where the dimensions, the tolerances and even the architecture are not finalized, will be the definitive one. Therefore, the optimization phase will be presented as a two-step process: first, a more flexible optimization will be described, suitable for an initial analysis and capable of identifying multiple solutions, and then a more stringent optimization, aimed at finding the best possible design. Such optimization process will be applied to a relevant case study, with the addition of multi-body simulations to reproduce the dynamic behavior of the balanced crankshaft, that will help in verifying the capabilities of the methodology and identifying the pros and cons of different optimization algorithms.

2. Methodology

The crankshaft of an internal combustion engine transmits the translational movement generated by the pistons to the wheels in the form of a rotary motion, thus allowing the movement of the vehicle. The complex kinematic chain that characterizes the crankshaft involves many forces. The inertial translational forces are due to the translating masses, which include the pistons and a part of the connecting rods, while the centrifugal forces are due to the rotating masses, which include most of the connecting rods and their pins. Such forces, given their cyclic nature, create deformations of considerable importance on the shaft, which should maintain a perfect rotation around its axis. Due to these deformations, the shaft may incur in a reduced operating life and vibrations of high intensity may be transmitted to the entire vehicle, reducing the comfort of the occupants.

2.1. Counterweight definition

In order to reduce deformations, the forces and moments acting on the crankshaft must be countered with equal and opposite equivalents. In the case of centrifugal forces, the main topic of this memory, each crank generates a force equal to:

$$F_\omega = m_{rot}\,\omega^2\,r \tag{1}$$

where m_{rot} is the rotating mass, given by the sum of the rotating masses of the connecting rod, connecting rod pins and cranks, ω the speed of the motor, r the crank radius.

This force is always equal in absolute value while the direction changes continuously. The best way to counteract these forces is to apply balancing counterweights, as shown in Fig. 1.

Counterweights are placed so that their mass, having a center of gravity at a certain distance from the axis of rotation, generates a resulting centrifugal force equal in amplitude and opposite in direction to that of the shaft itself. The centrifugal force of each counterweight can thus be expressed as:

$$F_{cw} = m_{cw}\,\omega^2\,y_{cw} \tag{2}$$

where m_{cw} is the mass of the counterweight and y_{cw} the distance of its center of gravity to the axis of rotation of the shaft.

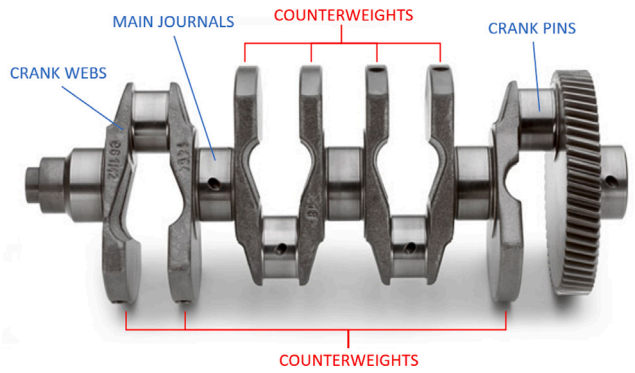


Fig. 1. Crankshaft example [26], with corresponding parts.

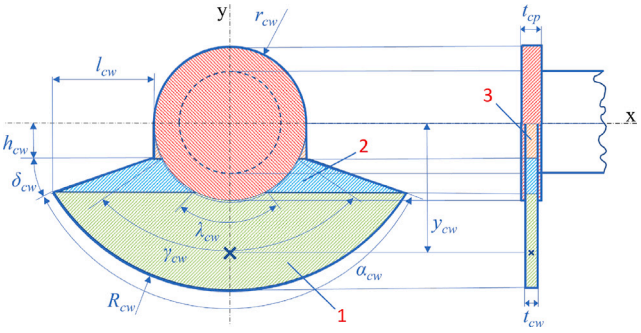


Fig. 2. Geometric characterization of a generic counterweight.

Since the rotational velocity ω of the shaft is fixed, the designer may act on the mass m_{cw} and the center of gravity position y_{cw}. In order to achieve this goal, it is necessary, first of all, to characterize the geometry of the counterweight section. Given its complex shape, it is convenient to divide the cross-section into elementary geometric areas, as shown in Fig. 2.

The area of each section is derived from the following equations:

$$A_1 = R_{cw}^2\left(\frac{\alpha_{cw}}{2} - \sin\frac{\alpha_{cw}}{2}\cos\frac{\alpha_{cw}}{2}\right) - r_{cw}^2\left(\frac{\lambda_{cw}}{2} - \sin\frac{\lambda_{cw}}{2}\cos\frac{\lambda_{cw}}{2}\right) \tag{3}$$

$$A_2 = (2r_{cw} + l_{cw})|l_{cw}|\tan\delta_{cw} - r_{cw}^2\left(\frac{\gamma_{cw}}{2} - \sin\frac{\gamma_{cw}}{2}\cos\frac{\gamma_{cw}}{2}\right) + r_{cw}^2\left(\frac{\lambda_{cw}}{2} - \sin\frac{\lambda_{cw}}{2}\cos\frac{\lambda_{cw}}{2}\right) \tag{4}$$

$$A_3 = 2r_{cw}h_{cw} - r_{cw}^2\frac{\pi}{2} + r_{cw}^2\left(\frac{\gamma_{cw}}{2} - \sin\frac{\gamma_{cw}}{2}\cos\frac{\gamma_{cw}}{2}\right) \tag{5}$$

$$l_{cw} = \frac{2R_{cw}\sin\frac{\alpha_{cw}}{2} - 2r_{cw}}{2} \tag{6}$$

where the different geometric quantities are indicated in Fig. 2.

The area highlighted in red is a constituent part of the crankshaft and is therefore considered as an input of the optimization, as a result r_{cw} and t_{cp} may be considered as project parameters. Instead, l_{cw}, derived via Eq. (6), is an additional parameter that helps define the geometry of the counterweight. From the values obtained from the Eq. (3), (4) and (5), the total area of the counterweight cross-section can be expressed as:

$$A_{cw} = A_1 + A_2 + A_3 \tag{7}$$

from which its mass is obtained:

$$m_{cw} = A_{cw}t_{cw}\rho \tag{8}$$

where ρ is the density of the counterweight.

Once the value of the areas has been calculated, it is then possible to determine the position of the center of gravity of each area:

$$y_1 = \frac{1}{A_1}\left\{R_{cw}^2\left(\frac{\alpha_{cw}}{2} - \sin\frac{\alpha_{cw}}{2}\cos\frac{\alpha_{cw}}{2}\right)\left[h_{cw} + |l_{cw}|\tan\delta_{cw} - R_{cw}\cos\frac{\alpha_{cw}}{2} + \frac{\left(2R_{cw}\sin\frac{\alpha_{cw}}{2}\right)^3}{12R_{cw}^2\left(\frac{\alpha_{cw}}{2} - \sin\frac{\alpha_{cw}}{2}\cos\frac{\alpha_{cw}}{2}\right)}\right] - \left[\frac{2}{3}r_{cw}^3\left(\sin\frac{\lambda_{cw}}{2}\right)^3\right]\right\} \quad (9)$$

$$y_2 = \frac{1}{A_2}\left\{(2r_{cw} + l_c)|l_{cw}|\tan\delta_{cw}\left[h_{cw} + |l_{cw}|\tan\delta_{cw} - \frac{|l_{cw}|\tan\delta_{cw}(3r_{cw} + l_{cw})}{3(2r_{cw} + l_{cw})}\right] - \left[\frac{2}{3}r_{cw}^3\left(\sin\frac{\gamma_{cw}}{2}\right)^3 - \left(\sin\frac{\lambda_{cw}}{2}\right)^3\right]\right\} \quad (10)$$

$$y_3 = \frac{\left(h_{cw}^2 r_{cw}\right) - \left[r_{cw}^3\left(\frac{2}{3} - \frac{2}{3}(\sin\frac{\gamma_{cw}}{2})^3\right)\right]}{A_3} \quad (11)$$

from which it is then possible to evaluate the position of the center of gravity of the whole counterweight:

$$y_{cw} = \frac{y_1 A_1 + y_2 A_2 + y_3 A_3}{A_1 + A_2 + A_3} \quad (12)$$

Finally, it is possible to derive the two angular parameters λ and γ that appear within the equations from (3) to (12):

$$\begin{cases} \lambda_{cw} = 2\cos^{-1}\left(\frac{h_R + |l_{cw}|\tan\delta_{cw}}{r_{cw}}\right) & \text{if } h_{cw} + |l_{cw}|\tan(\delta_{cw}) \le r_{cw} \\ \lambda_{cw} = 0 & \text{if } h_{cw} + |l_{cw}|\tan(\delta_{cw}) > r_{cw} \end{cases} \quad (13)$$

$$\begin{cases} \gamma_{cw} = 2\cos^{-1}\left(\frac{h_{cw}}{r_{cw}}\right) & \text{if } h_{cw} \le r_{cw} \\ \gamma_{cw} = 0 & \text{if } h_{cw} > r_{cw} \end{cases} \quad (14)$$

From the Eqs. (3) to (14), it is clear that the independent parameters are only 5: R_{cw}, h_{cw}, t_{cw}, α_{cw} and δ_{cw}. These, therefore, are the only parameters that an optimization algorithm should modify in order to find the best geometry for a given crankshaft configuration. These equations have, however, been implemented in a MATLAB® script, which calculates all the remaining variables for each set of independent parameters.

2.2. Optimization setup

The fact that a crankshaft, and more generally an internal combustion engine, can have very different configurations and geometries makes it very difficult to find an optimal solution for balancing. In fact, as the number of cylinders, their arrangement, the order of ignition and the geometric parameters vary, the configuration and the number of counterweights to be applied also change. A MATLAB® tool, dedicated to the evaluation of the optimal counterweight to balance different crankshaft configurations has been presented in [27,28]. This tool is based on a matrix approach that identifies where and how to place counterweights on the crankshaft and calculates the individual force and moment contributions of all rotating and translating elements.

The tool evaluates how a certain geometry of the counterweights behaves with a view to balancing the crankshaft, automatically carrying out all the necessary calculations. These calculations include the evaluation of the maximum radial size of the counterweight, which allows the optimized geometry to respect the dimensions dictated by the architecture of the internal combustion engine, and the balancing percentages, both for forces and for moments. These percentages are

```
environment
    tabular_data
        tabular_data_file "tabular.data"
    output_file "dakota.out"
    error_file "dakota.err"
    write_restart "dakota.rst"
    results_output
        hdf5

method
    id_method "method1"
    moga
        model_pointer "model1"
        max_iterations = 150
        max_function_evaluations = 5000
        convergence_tolerance = 0.1

model
    id_model "model1"
    single
        interface_pointer "interface1"
    variables_pointer "variables1"
    responses_pointer "responses1"

variables
    id_variables "variables1"
    active
        design
    continuous_design 5
        initial_point 100 25 7.5 90 22.5
        lower_bounds 0   0 5 0 0
        upper_bounds 200 50 12.6 180 45
        descriptors "R_cw" "h_cw" "t_cw" "alpha_cw" "delta_cw"

interface
    id_interface "interface1"
    analysis_drivers "CW_driver.py"
      parameters_file = 'params.in'
      results_file    = 'results.out'
        system
            file_save
    asynchronous
    work_directory directory_tag directory_save named 'workdir'
    copy_files = 'templatedir/*'
    evaluation_concurrency = 4

responses
    id_responses "responses1"
    descriptors "m_cw" "R_cw_min" "K_bilF" "K_bilM"
    objective_functions 2
    sense "minimize"
    nonlinear_inequality_constraints 2
        lower_bounds 90 90
        upper_bounds 100 100
    no_gradients
    no_hessians
```

Fig. 3. Dakota file for setting up parsing.

fundamental in the optimization process as they assess how effective a certain counterweight configuration is in balancing the shaft, without imposing a fixed balancing performance that would be difficult for an algorithm to respect.

Once the numerical simulation of the counterweight geometry has been defined, the next step involves the formulation of the optimization. This phase was implemented through the open-source Dakota software, developed by Sandia Laboratories, which allows to carry out a multitude of analyses, including parametric studies, Design of Experiments and optimizations. The latter is the feature that has been deepened here.

The logic of operation of Dakota is quite simple: it is based on the creation of a file that defines the settings of the analysis, of which an example is given in Fig. 3.

It can be noted that the file is divided into six sections, each dedicated to a certain function; the most relevant for the analysis are:

- method, defines the nature of the analysis to be carried out, the solver and its parameters;
- variables, defines the type and numerical limits of each variable involved in the analysis;

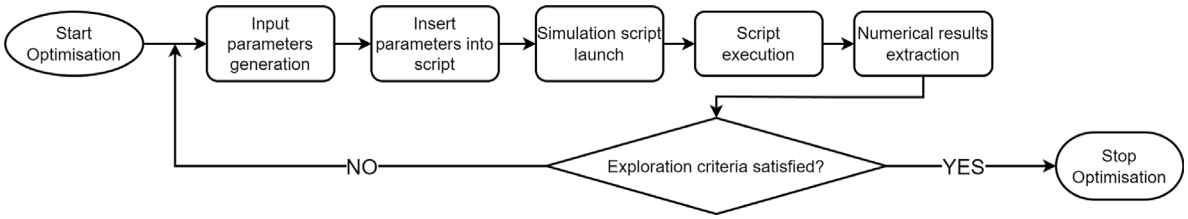


Fig. 4. Optimization loop followed by Dakota.

Table 1
Analysis input parameters with their respective limit values.

Parameter	R_{cw} [mm]	h_{cw} [mm]	t_{cw} [mm]	α_{cw} [deg]	δ_{cw} [deg]
MIN	0	0	5	0	0
MAX	200	50	12.6	180	45

Table 2
Constraint variables with their respective limit values.

Parameter	$R_{cw,min}$ [mm]	$K_{bil,F}$ [%]	$K_{bil,M}$ [%]
MIN	0	90	90
MAX	75	100	100

- interface, defines the files useful for the analysis, including the "driver", which acts as a link between Dakota and the simulation code;
- responses, specifies response variables, including any constraints.

Particular attention must be paid to the definition of the "driver". This file, in fact, is directly responsible for the pre-processing, simulation code launch and post-processing phases. The pre-processing phase consists in replacing the input parameters (in the case presented here, the 5 parameters previously indicated) with values generated by Dakota, based on the constraints imposed and the analysis settings defined by the user, within the simulation code. The file will continue to launch the code itself, and then will retrieve and store the simulation results for a certain set of inputs. Finally, the post-processing will collect the results in a file readable by Dakota, used to assess whether the study is proceeding and if the results are respecting the constraints imposed. The driver for counterweight optimization consists of a Python script, which, using well-defined commands, is able to modify the original MATLAB® code, launch it, and collect the results. Fig. 4 shows a representative diagram of this process.

It is possible to notice that different stages of optimization are handled by different files, with the Dakota input file acting as a "master".It is also possible to observe how the optimization is regulated by "Exploration Criteria": these are parameters that allow Dakota to evaluate the goodness of the analysis. The primary criterion is optimization convergence, which is different for individual algorithms, but a limit on the number of iterations the code can perform may also be set.

Once all these aspects of optimization are established, the input file must be prepared so that Dakota can perform the desired analysis. First, it is crucial to indicate which optimization algorithm to employ, many of which are already native to Dakota. Due to its importance, this topic will be discussed later. The input parameters and the maximum and minimum values for each of them were then defined, as reported in Table 1. Similarly, the limits for the constraint variables, reported in Table 2, may be defined according to prior knowledge on the case study.

The variables $K_{bil,F}$ and $K_{bil,M}$ are defined as the balancing percentages of forces and moments respectively. They are derived by the following equations:

$$K_{bil,F} = \frac{F_{cw,res}}{F_{cs,res}} \cdot 100 \tag{15}$$

$$K_{bil,M} = \frac{M_{cw,res}}{M_{cs,res}} \cdot 100 \tag{16}$$

where $F_{cw,res}$ and $F_{cs,res}$ are the resulting force of the counterweights and crankshaft, respectively, found as the sum of their respective x and z components, as follows:

$$F_{cw,res} = \sqrt{F_{cw,x}^2 + F_{cw,z}^2} \qquad F_{cs,res} = \sqrt{F_{cs,x}^2 + F_{cs,z}^2} \tag{17}$$

where:

$$\begin{cases} F_{cw,x} = \sum_{j=1}^{2N_{cr}} m_{cw}\omega^2 y_{cw} CW(1,j) \\ F_{cw,z} = \sum_{j=1}^{2N_{cr}} m_{cw}\omega^2 y_{cw} CW(3,j) \end{cases} \tag{18}$$

$$\begin{cases} F_{cs,x} = \sum_{i=1}^{N_{cr}} m_{rot}\omega^2 rCS(1,i) \\ F_{cs,z} = \sum_{i=1}^{N_{cr}} m_{rot}\omega^2 rCS(3,i) \end{cases} \tag{19}$$

while $M_{cw,res}$ and $M_{cs,res}$ are the resulting moment of the counterweights and the crankshaft, respectively, found in a similar manner:

$$M_{cw,res} = \sqrt{M_{cw,x}^2 + M_{cw,z}^2} \qquad M_{cs,res} = \sqrt{M_{cs,x}^2 + M_{cs,z}^2} \tag{20}$$

where:

$$\begin{cases} M_{cw,x} = \sum_{j=1}^{2N_{cr}} m_{cw}\omega^2 y_{cw} CW(1,j) CW(2,j) \\ M_{cw,z} = \sum_{j=1}^{2N_{cr}} m_{cw}\omega^2 y_{cw} CW(3,j) CW(2,j) \end{cases} \tag{21}$$

$$\begin{cases} M_{cs,x} = \sum_{i=1}^{N_{cr}} m_{rot}\omega^2 rCS(1,i) CS(2,i) \\ M_{cs,z} = \sum_{i=1}^{N_{cr}} m_{rot}\omega^2 rCS(3,i) CS(2,i) \end{cases} \tag{22}$$

In these equations, two terms can be seen, CS and CW: they represent the orientation matrices of the cranks and the counterweight, respectively, of the crankshaft. They are composed by 3-component vectors, one for each crank/counterweight, defined as follows:

$$\begin{cases} CS(1,N_{fo}) = [sin(\theta + (i-1)\phi)] \\ CS(2,N_{fo}) = \left[(N_{fo}-1)a - \frac{(N_{cr}-1)}{2}a\right] \\ CS(3,N_{fo}) = [cos(\theta + (i-1)\phi)] \end{cases} \tag{23}$$

$$\begin{cases} CW(1,j) = [sin(\theta + (i-1)\phi + \alpha)](-N_{cw}) \\ CW(2,j) = \left[\left[(N_{fo}-1)a - \frac{(N_{cr}-1)}{2}a\right] + \tau\right](N_{cw}) \\ CW(3,j) = [cos(\theta + (i-1)\phi + \alpha)](-N_{cw}) \end{cases} \tag{24}$$

where $\tau = a/4$ for $j = 2i$ and $\tau = -a/4$ for $j = 2i-1$, a is the distance between cylinders axis, θ is the crank angle, ϕ is the phase between cranks, N_{fo} is the cylinder number based on the firing order, α is the angle offset between counterweight and crankpin, N_{cr} is the number of cranks, N_{cw} is equal to 1 if the crank-web has a counterweight, otherwise is zero Finally, $1 \le i \le N_{cr} \in \mathbb{N}$ and $\forall i\ N_{cw} = Y_{cw}(j)$ with $j = 2i-1 \wedge j = 2i$.

It can be noted that the values imposed for the balancing percentages of forces ($K_{bil,F}$) and moments ($K_{bil,M}$) are between 90% and 100%: this will lead to a close-to-ideal shaft balance, allowing acceptable algorithm calculation times. If a value of 100% were to be

imposed, as already mentioned, the algorithm would have considerable difficulty converging to a solution. The maximum radial dimension $R_{cw,min}$, on the other hand, has been imposed to avoid interpenetration with the pistons. It is defined as:

$$R_{cw,min} = MAX(R^{top}_{cw,min}, R^{side}_{cw,min}) \tag{25}$$

where $R^{top}_{cw,min}$ and $R^{side}_{cw,min}$ are derived from:

$$R^{top}_{cw,min} = h_{cw} + |l_{cw}| \tan(\delta_{cw}) + \left[R_{cw}\left(1 - \cos\frac{\alpha_{cw}}{2}\right)\right] \tag{26}$$

$$R^{side}_{cw,min} = \sqrt{(l_{cw} + r_{cw})^2 + (h_{cw} + |l_{cw}| \tan\delta_{cw})^2} \tag{27}$$

Another important information to be specified in the Dakota file is the actual objective of the analysis. As a general rule, the most common objectives in the counterweight design is to minimize its mass and/or its radial dimension. Therefore, two possible optimization problems can be defined:

$$min\left[m_{cw}(R_{cw}, h_{cw}, t_{cw}, \alpha_{cw}, \delta_{cw})\right] \tag{28}$$

$$min\left[m_{cw}(R_{cw}, h_{cw}, t_{cw}, \alpha_{cw}, \delta_{cw}), R_{cw,min}(R_{cw}, h_{cw}, t_{cw}, \alpha_{cw}, \delta_{cw})\right] \tag{29}$$

The problem in Eq. (28) is a single objective problem, where the mass must be minimized, meanwhile the one in Eq. (29) is a multi-objective problem, where both the mass and the radial dimension need to be minimized; as such, in this optimization $R_{cw,min}$ is not a constraint variable as described in Table 2.

The last step in setting up the analysis is to define the configuration of the crankshaft that must be balanced, inserting in the MATLAB® script the geometric and mass parameters of the crankshaft and the other components of the crank mechanism.

2.3. Choice of algorithm and approach

As already stated, the design of an engine can go through multiple phases before the final solution is confirmed. Therefore, it may be inefficient to perform a full optimization, which is generally time-consuming, at the very beginning of the design process: in fact, if an optimized counterweight design is found for a specific early engine configuration, and then such configuration is modified, the counterweight may become incompatible with the current engine, wasting development time and costs.

Therefore, the approach here proposed is a two-step optimization, which can help in reducing the discarded incompatible solutions and improving the flexibility of the design process. The first step involves the use of multi-objective algorithms, which can be used with an early configuration of the engine to find a wider range of possible 'good-enough' solutions, thanks to the identification of the Pareto front [29]. The Pareto front, in fact, identifies a series of best possible solutions based on two objective functions, based on the "weight" assigned to each one. Thanks to this "front" it is possible to be more flexible in the counterweight design, since, in case of changes in the engine architecture or dimensions, another compatible solution on the Pareto front can be readily identified. To this purpose, two algorithms will be showcased:

- MOGA (Multi-Objective Genetic Algorithm) [30];
- PARETO SET [31] with SOGA.

The second step, instead, is a more detail-centered optimization. With this step, the goal is no longer to have a portfolio of viable solutions to allow for flexibility, but to find the best possible solution for a certain engine configuration. For this reason, it is important that this step is carried out only when the configuration is being finalized, since this optimization uses the final constraints of the architecture to minimize a specific counterweight parameter, usually its mass, that satisfies the requirements of that engine. For this phase, three single-objective algorithms were tested:

- SOGA (Single-Objective Genetic Algorithm) [32];
- COLINY DIRECT [33];
- PATTERN SEARCH [34].

The two-step optimization procedure is graphically represented in Fig. 5. In order to test the approach, a two-cylinder in-line engine with a crank angle of 360° was chosen a case study. Fig. 6 shows the CAD of the entire unbalanced crank used for this analysis.

Table 3
Numerical results of the extreme and best trade-off solutions obtained through the MOGA optimization algorithm.

	Best m_{cw}	Best $R_{cw,min}$	Best trade-off
R_{cw} [mm]	61.32	76.09	91.96
h_{cw} [mm]	47.16	28.10	47.16
t_{cw} [mm]	6.76	10.53	8.98
α_{cw} [deg]	125.46	127.37	84.99
δ_{cw} [deg]	41.26	9.22	25.06
m_{cw} [g]	309.7	415.6	345.3
$R_{cw,min}$ [mm]	100.48	76.40	87.37
$K_{bil,F}$ [%]	90.39	92.30	90.15
$K_{bil,M}$ [%]	100	100	100

3. Results

With the optimization process laid out, it is now possible to proceed with the execution of the counterweight optimization, performing the multi-objective analysis first and then the single-objective one. Each optimization was carried out on a laptop PC with 8 core 2.3 GHz CPU and 16 GB of 3200 MHz RAM. The ρ density of the material was set at 8000 kg/m^3 for each simulation performed.

3.1. Multi-objective optimization

As previously introduced, the first step in designing a counterweight should be to identify multiple possible solutions, starting from the initial constraints imposed by the other engine components, in order to be more flexible in the design process. In this chapter the MOGA algorithm and then the PARETO SET will be show-cased.

The MOGA algorithm (or Multi-Objective Genetic Algorithm) is the first method employed to search for the best solutions of the previously presented problem. In basic terms, MOGA is a genetic algorithm, meaning that at every iteration it will create a new set of parameters based on the results of the previous one [30,35]. Being a multi-objective algorithm, the final results of the optimization will not be univocal, as can be seen in Fig. 7, since it depends on the bias that each solution has with respect to each objective function. As a matter of fact, for the current optimization problem the best solutions found were 17. For sake of brevity, in Fig. 8 only the geometries of the two "extremes" and the best trade-off solutions are shown. Additionally, the numerical results are reported in Table 3.

The geometric characteristics of the three counterweights are markedly different: the one in Fig. 8(a) sacrifices the radial dimension to reduce the mass of the counterweight, while the one in Fig. 8(b) is the other way around. Finally, the best trade-off solution shown in Fig. 8(c) represents a compromise between the two "extremes".

As previously mentioned, a multi-objective optimization can give the designer a wider view of the problem, allowing higher adaptability functional to the ever-changing early design phase. Such flexibility can be visualized through the Pareto front, which shows the trade-off trend between the objective functions. Said front for the MOGA optimization is shown in Fig. 9, derived from the results as reported

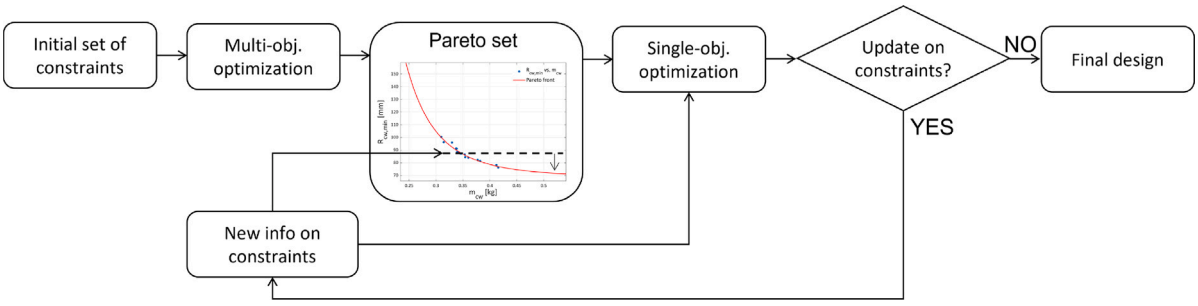


Fig. 5. Flowchart representing the two-step optimization procedure.

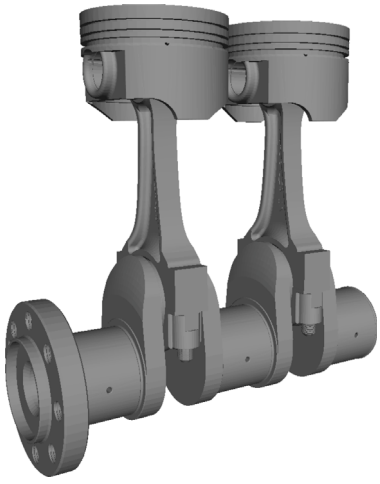


Fig. 6. CAD model of the entire in-line 2-cylinder crank used as a case study, without counterweights.

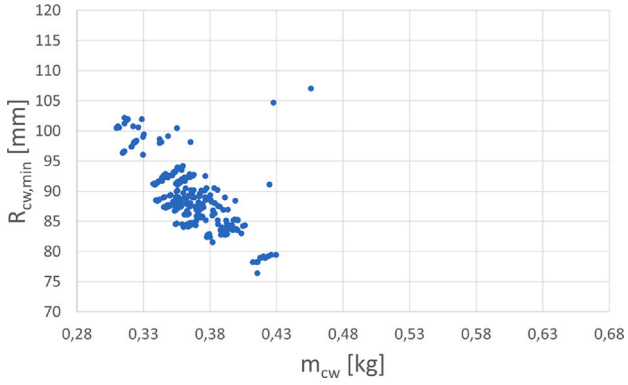
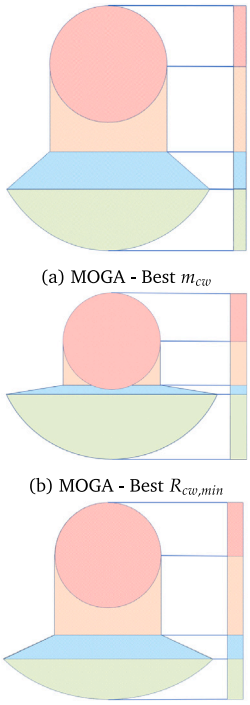


Fig. 7. Plot of the radial dimension $R_{cw,min}$ with respect to the mass for the solutions of the MOGA algorithm.



(a) MOGA - Best m_{cw}

(b) MOGA - Best $R_{cw,min}$

(c) MOGA - Best trade-off

Fig. 8. Geometries of counterweights obtained through the MOGA algorithm.

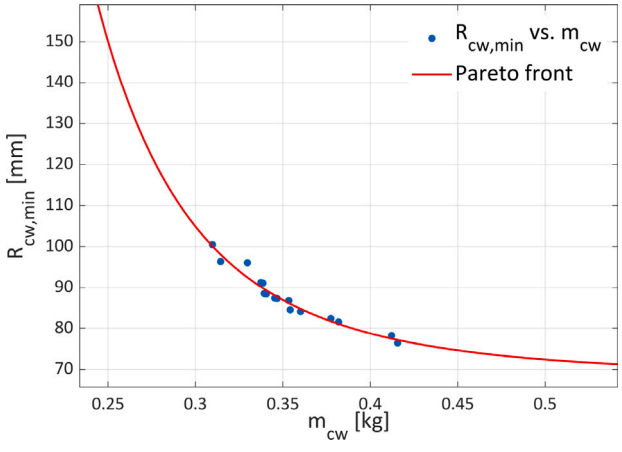
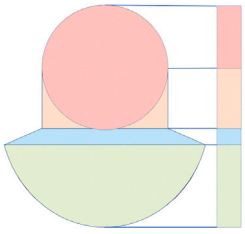


Fig. 9. Pareto front of the best solutions found through the MOGA algorithm.

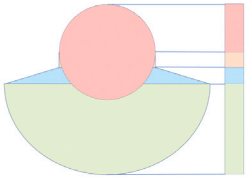
in Fig. 7. The front immediately shows the trend of the objective functions with respect to each other, demonstrating that a smaller mass of the counterweight requires a bigger radial dimension and vice versa. Therefore, if the engine requirements change during development, the designer can readily make decisions to accommodate them. An important aspect to keep into consideration is with regards to the balancing percentages: in fact, in all simulations, including the ones that will be described later, $K_{bil,M} = 100\%$ has been obtained. This is due to the natural balance of a 2-cylinder in-line engine with a 360° phase with respect to centrifugal moments. Meanwhile, $K_{bil,F}$ is always around 90%, due to the constraint on it imposed during the optimizations setup, as explained earlier, causing a tendency to maintain it on near the minimum value.

Table 4
Weights assigned to the objective functions for the PARETO SET optimization.

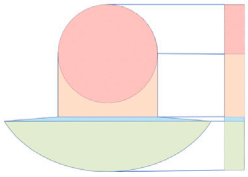
	$R_{cw,min}$	m_{cw}
Set 1	0.5	0.5
Set 2	0.75	0.25
Set 3	0.25	0.75



(a) PARETO SET - Best m_{cw}



(b) PARETO SET - Best $R_{cw,min}$



(c) PARETO SET - Best trade-off

Fig. 10. Geometries of counterweights obtained through the PARETO SET optimization strategy.

Table 5
Numerical results of the three solutions obtained through the PARETO SET optimization strategy.

	Best m_{cw}	Best $R_{cw,min}$	Best trade-off
R_{cw} [mm]	50.81	62.07	82.07
h_{cw} [mm]	30.70	10.16	39.91
t_{cw} [mm]	11.96	12.11	12.05
α_{cw} [deg]	157.82	165.04	104.42
δ_{cw} [deg]	24.09	16.95	4.82
m_{cw} [g]	392.8	482.0	395.1
$R_{cw,min}$ [mm]	79.91	73.27	77.66
$K_{bil,F}$ [%]	90.89	93.45	90.02
$K_{bil,M}$ [%]	100	100	100

The second multi-objective optimization strategy that was tested was the PARETO SET optimization. This method differs quite a lot form the previous MOGA algorithm: in fact, the PARETO SET performs multiple optimization runs, each with different weights assigned to the objective functions. It then employs a single-objective algorithm to find the optimal solution, based on the assigned weights. Therefore, an optimal solution will be found for each set of weights [36].

In the current contribution, the SOGA or Single-objective Genetic Algorithm was used as the single-objective algorithm, which follows the same logic as the MOGA but for only one objective function. in order to contain the optimization time, it was decided to use three different sets of weights for the two objective function, as reported in Table 4.

From the optimization, the three best solutions were found; in Fig. 10 their respective geometries are reported, while in Table 5 the numerical results can be seen.

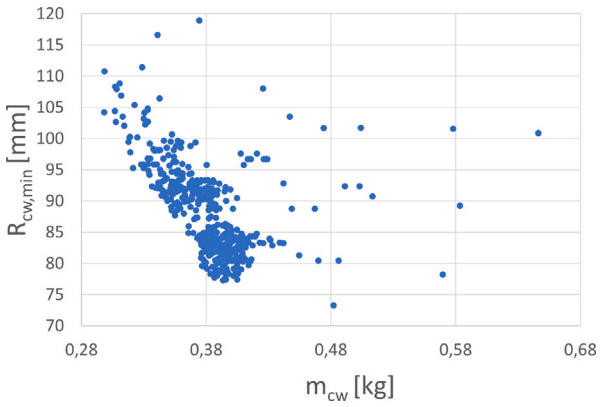


Fig. 11. Plot of the radial dimension $R_{cw,min}$ with respect to the mass for the solutions of the PARETO SET strategy.

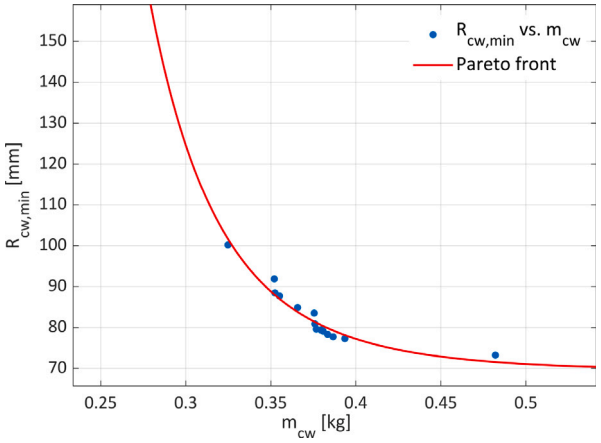


Fig. 12. Pareto front of the best solutions found through the PARETO SET optimization strategy.

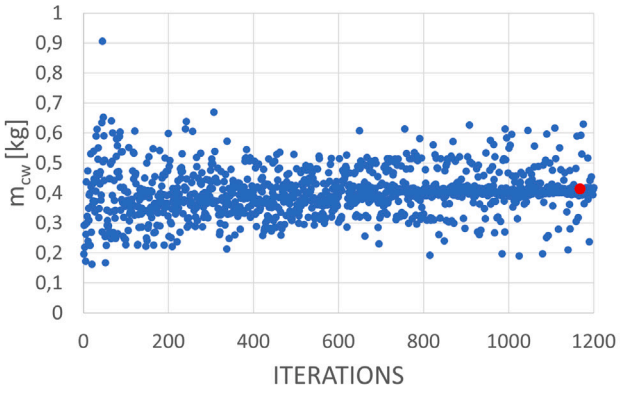
We can notice a similar trend to the geometries obtained through the MOGA algorithm, further confirming the relation between mass and radial dimension of the counterweights. In this case, however, due to the low number of optimal solutions found, the Pareto front would not provide a reliable estimation of the trade-off between objective functions. In order to circumvent this issue, it is possible to consider all the solutions found by the optimization and therefore capture the front from a much bigger set of data. In fact, if the plot $R_{cw,min}$ vs. m_{cw} of all the found solutions is created, as visible in Fig. 11, a Pareto front is easily identifiable. Hence, through a dedicated function for the identification of the Pareto front data points, it was possible to generate the front for the data obtained from the PARETO SET strategy, as showcased in Fig. 12.

3.2. Single-objective optimization

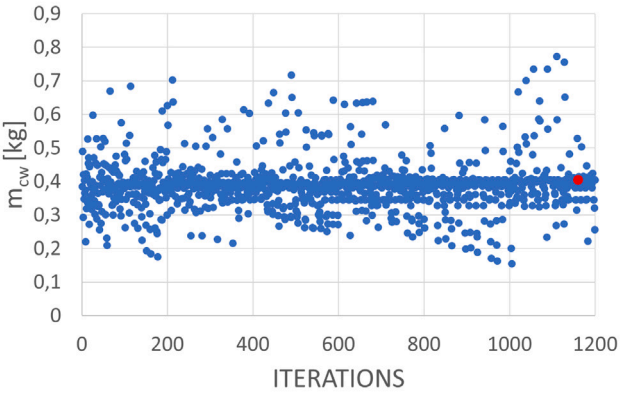
Once the multi-objective optimization has been completed, the next natural step is to perform a single-objective optimization in order to find the best possible solution for the specific requirements of the engine development. For this contribution, the optimization of the counterweight mass has been performed, employing the three algorithm previously introduced. As it will be showcased later, the MOGA algorithm provided the overall better results in terms of quality. Therefore, the results of this optimization has been used as the starting point for the single objective optimization. In particular, the single-objective

Table 6
Limits on the input geometrical parameters for the single-objective optimization found form the MOGA optimization.

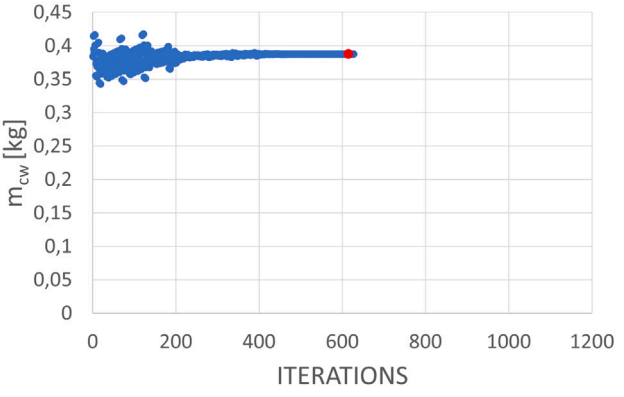
Parameter	R_{cw} [mm]	h_{cw} [mm]	t_{cw} [mm]	α_{cw} [deg]	δ_{cw} [deg]
MIN	62	28	6	84	8
MAX	92	49	11	128	42



(a) SOGA



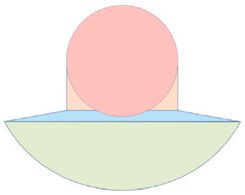
(b) COLINY DIRECT



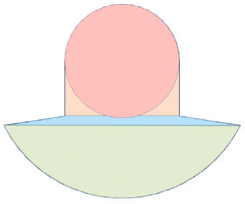
(c) PATTERN SEARCH

Fig. 13. The solutions found by the three optimization algorithms, with the best solution found highlighted in red.

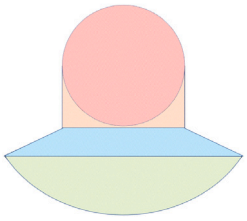
optimization has been setup with limits on the geometric parameters equal to the maximum and minimum values found with the multi-objective optimization (reported in Table 6) and with the constraint variables presented in Table 2. In Fig. 13 all the solutions found by the three algorithms are reported, while Fig. 14 shows the geometries of the individual counterweights obtained by each algorithm.



(a) SOGA



(b) COLINY DIRECT



(c) PATTERN SEARCH

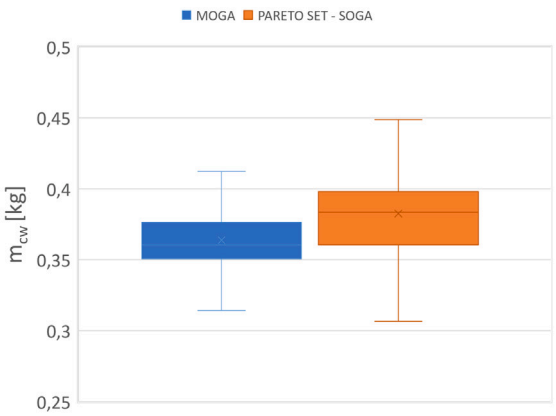
Fig. 14. Geometries of counterweights obtained through optimization algorithms.

Table 7
Numerical results obtained through the 3 single-objective optimization algorithms.

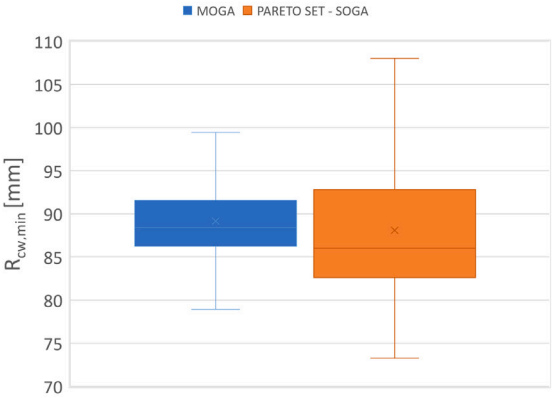
	SOGA	COLINY	PATTERN
R_{cw} [mm]	76.64	72.64	76.87
h_{cw} [mm]	28.79	30.33	32.20
t_{cw} [mm]	11.00	11.00	11.00
α_{cw} [deg]	122.13	127.26	106.13
δ_{cw} [deg]	10.08	8.44	25.11
m_{cw} [g]	408.8	403.9	387.5
$R_{cw,min}$ [mm]	75.70	75.68	76.87
$K_{bil,F}$ [%]	89.72	89.81	89.50
$K_{bil,M}$ [%]	100	100	100

It is noticeable how the geometries of the three single-objective algorithms (SOGA, COLINY DIRECT and PATTERN SEARCH) present very similar lateral geometry, confirming that all three algorithms, each following their own logic, converge on a quasi-univocal solution, demonstrating that, for a defined crankshaft and a certain set of constraints, the overall optimal solution is only one. Table 7 shows the numerical results of the 3 optimizations, which were used to create the corresponding sections of the counterweights. The only noticeable difference is in the solution found by the PATTERN SEARCH algorithm, which finds a lower value of the counterweight mass with a slight variation of the geometry. However, as it can be noticed in the numerical results, this comes at the cost of a lower respect of the constraint on the radial dimension.

On this aspect, it is noticeable that all three solutions tend to obtain a radial dimension very slightly higher of the maximum imposed value. This shows that the logic implemented by the three algorithm is setup in a way that allows for a margin of error on the constraint, so that, if the found value respects it, convergence is still reached. Additionally, very importantly, it demonstrates that, although convergence is reached, the constraints were actually very harsh, since, in order to respect the balancing constraints, the radial dimension constraint was not mathematically respected. In fact, if the plot in Fig. 9 is again considered, it can be seen that $R_{cw,min} = 75$ mm is achievable on the far-right section of the Pareto front, inevitably rendering the identification



(a) Boxplot for m_{cw}



(b) Boxplot for $R_{cw,min}$

Fig. 15. Boxplots for the two objective functions for both multi-objective algorithms.

of a suitable solution quite harder on the algorithm. In any case, the output of this phase can be considered as the one that could be employed on the final configuration of the engine, since it is the most optimal solution for that specific architecture.

3.3. Algorithms comparison

In the present work, multiple algorithms were employed in order to obtain the best counterweight geometry for given engine constraints. Each one has different characteristics, pros and cons, making each of them more suited to a certain application than another. Therefore, it is very interesting to draw a comparison between them. One approach that could be used for this task is the generation of the Boxplots for each data set of the algorithms.

First, in Fig. 15 the Boxplots for the two objective functions for both multi-objective algorithm employed are reported. In the case of the PARETO SET plot, only the results for the best trade-off run are plotted. Meanwhile, in Fig. 16, the Boxplot of the three single-objective algorithms results is reported. From these plots a few considerations can be made. First of all, it is noticeable that by comparing the two multi-objective algorithm the MOGA is overall more accurate in finding the best solutions, since the dispersion of the data is narrower. Additionally, the MOGA was capable of reaching convergence around 40% faster, as it can be seen in Table 8. This is due to the fact that the PARETO SET effectively performs a new optimization for each set of weights, while MOGA needs to reach convergence only once.

Another interesting point is found by comparing the Boxplots regarding the found mass of the counterweight. It is clear that the search for the optimal solution is much more detailed in the single-objective

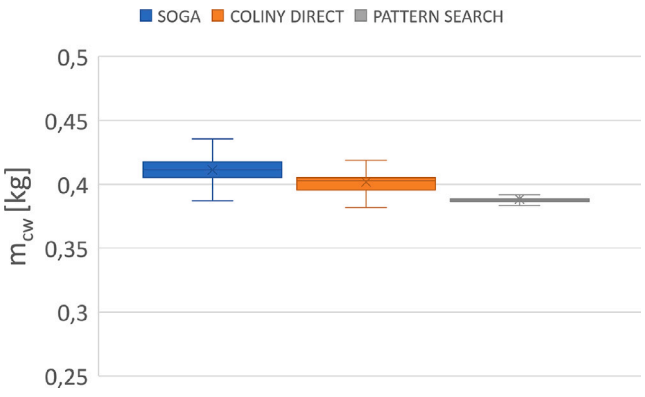


Fig. 16. Boxplot for the results of the three single-objective algorithms.

Table 8
Optimization times for the multi and single-objective algorithms.

Algorithm	Time [min]
MOGA	117
PARETO SET	187
SOGA	65
COLINY DIRECT	66
PATTERN SEARCH	26

algorithms, confirming that the multi-objective optimizations are more suited to a earlier design phase, granting the designer the possibility to have a wider view of the problem. Meanwhile, the single-objective algorithms are more useful in a more detail-oriented development phase.

It is immediately clear how, between the single-objective algorithms the one that yields the better results in terms of precision, showing a much lower dispersion of the results, is the PATTERN SEARCH algorithm. However, considering the set constraints, it is also the algorithm that respects them with a lower stringency, obtaining a overall radial dimension higher than the imposed limit. The other two algorithms are basically equivalent in terms of results, with the COLINY DIRECT algorithm showing a slightly lower dispersion of the solutions. A final consideration is relative to the time required to perform each optimization. As it can be noticed in Table 8, the multi-objective optimization tend to have quite higher optimization times, especially the PARETO SET. This is due to the higher complexity of the optimization, requiring to find the best solutions for multiple variables, often inversely proportional to each other. Moreover, since this kind of optimization is performed in an earlier stage of the design, the constraints are much less stringent: as such, the algorithm has to search the best solutions in a much wider analysis space, unlike the single-objective ones, which use the already found results to narrow the search field. Among the single-objective optimizations, instead, the PATTERN SEARCH shows a quite lower computational time, potentially making it the best one for this kind of optimizations. However, the designer must be mindful whether the algorithm is capable of respecting the project requirements and adopt another one accordingly.

3.4. Validation of results

In order to verify that the counterweights obtained through optimization are able to correctly balance the crankshaft, a multi-body simulation was carried out in which the entire crank was simulated with and without counterweights. For the sake of brevity and given that the results obtained by the single-objective algorithms are similar to each other, only the simulation with counterweights obtained through the SOGA algorithm is reported here. In Fig. 17 the model used is

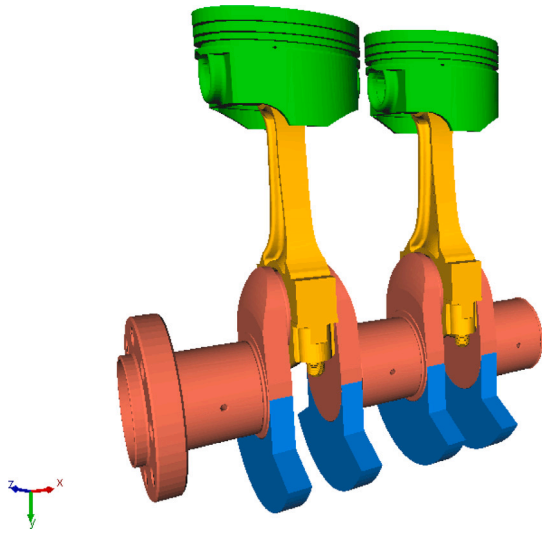
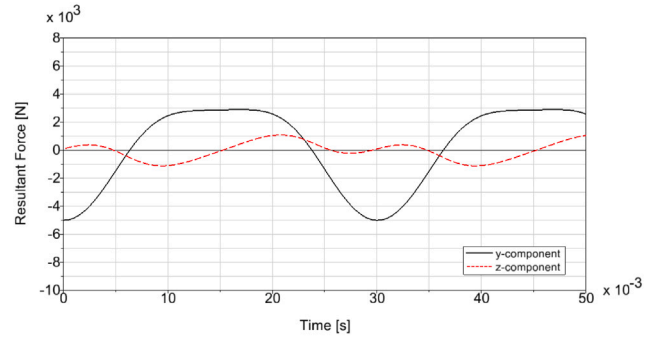
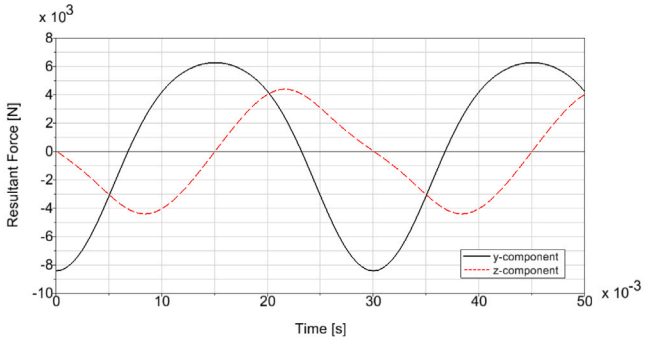


Fig. 17. CAD model of the entire crank with counterweights (in blue) obtained through the SOGA algorithm.



(a) With counterweights



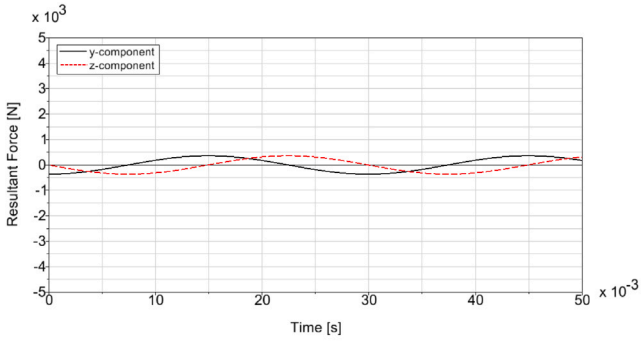
(b) Without counterweights

Fig. 18. Curves of the components of the resulting force, with and without counterweights.

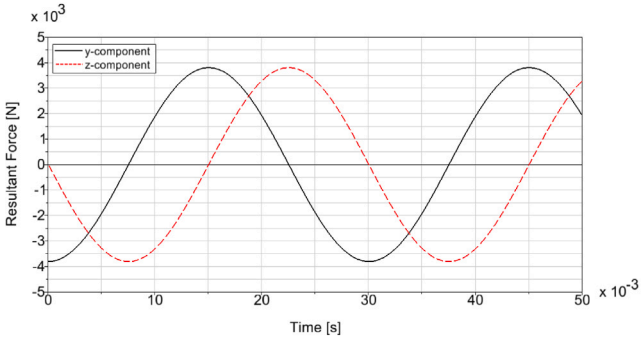
visible, with each component distinct and, in particular, in blue the counterweights obtained can be seen. The reference system used for the simulation is also reported (the axis of rotation of the shaft coincides with the x axis).

For the simulation, we set a shaft rotation speed of 2000 rpm, equal to the speed set for the MATLAB® code. Fig. 18 shows the curves of the components y and z of the resulting centrifugal force.

Comparing the curves of Fig. 18(a), where the counterweights were applied, with those of Fig. 18(b), where the crankshaft was left without counterweights (and therefore in an unbalanced condition), It can noticed the positive influence of the counterweights on the amplitude of



(a) With counterweights



(b) Without counterweights

Fig. 19. Curves of the components of the resulting force with the omission of pistons and connecting rods, with and without counterweights.

the curves, reducing the maximum values, and, in particular, allowing a "flattening" of the z component. However, given that the entire crank has been considered to generate these curves, a considerable contribution of force is given by the reciprocating inertial forces, which are not balanced by the counterweights of the crankshaft, thus making the counterweights effect less evident.

The simulation was therefore modified, removing the pistons and connecting rods (highlighted, respectively, in green and orange in Fig. 17) and adding the rotational mass of the connecting rods to the crank-pins. This way we obtain a crankshaft on which the translational components have no effect while taking into account all the rotational contributions of the crank, which must be balanced by counterweights. The result obtained is reported in Fig. 19 where it is noted, first of all, that the curves have become purely sinusoidal; this is perfectly in line with what could be expected as the rotational forces have constant absolute value and cyclically variable direction. It is therefore possible to confirm that those reported are purely centrifugal force curves.

Much more relevant is that it can be confirmed that the application of counterweights, and, above all, their optimized geometry, allows to correctly balance the crankshaft. In fact, it is found that the amplitude of the curves has gone from a value of about 3810 N, for both components, to a value of about 360 N, corresponding to a reduction of 90.53% compared to the unbalanced configuration. This reduction is in line with the balancing percentage obtained by the SOGA algorithm equal to 89.72% (reported in Table 7), with an error that can be considered insignificant.

4. Conclusion

The balancing of crankshafts for automotive applications is a complex matter that requires the designer to take into account different geometric and dynamic aspects at the same time. It is therefore becoming increasingly imperative to develop optimization methodologies that allow to obtain, in a short time and at low cost, geometries and

configurations of balancing systems that guarantee a high balance of the engine, keeping dimensions and masses contained.

To this end, the optimization procedure developed in this contribution allows to obtain counterweight geometries that meet these requirements, providing the designer with the opportunity to try numerous alternative configurations and obtain, for each, the optimal solution. This process is ensured by the tight integration of the two tools, MATLAB® and Dakota which, working in unison allow to effectively find the optimal overall solution.

Currently, the methodology proposed here is still at an early stage of development as optimization is guaranteed only for counterweights directly applied to the crankshaft. The next step will be to extend the optimization also to the balancing countershafts, which have the task of balancing the reciprocating inertial forces deriving from the rest of the crank. This will allow to have a complete optimization tool available to the designer, which will enable the achievement of the optimal configuration of the entire balancing system.

In parallel, it will also be necessary to extend the integration with other available tools, first of all a multi-body analysis tool, which allows automatic verification of the balancing performance configuration obtained. All this to guarantee a reduction in development times and the need for production of prototypes for experimental tests. Finally, a deepening of the optimizations techniques to adopt is certainly necessary, in order to guarantee a more comprehensive view on the whole optimization process and to find the best procedures and strategies to adopt based on the specific application requested by the designer.

CRediT authorship contribution statement

Eugenio Brusa: Conceptualization of this study, Proof-reading. **Alberto Dagna:** Conceptualization of this study, Methodology, Simulation, Writing. **Cristiana Delprete:** Conceptualization of this study, Methodology, Proof-reading. **Chiara Gastaldi:** Conceptualization of this study, Methodology, Proof-reading.

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.

References

[1] E. Brusa, D. Ferretto, A. Cala, Integration of heterogeneous functional-vs-physical simulation within the industrial system design activity, in: 2015 IEEE International Symposium on Systems Engineering, ISSE, IEEE, 2015, pp. 303–310, http://dx.doi.org/10.1109/syseng.2015.7302774.

[2] A.D. Nazarov, Compensating the total unbalanced mass of crankgear parts of a V-8 motor by changing the configuration of the crankshaft counterweights, Russian Engineering Research 27 (11) (2007) 738–744, http://dx.doi.org/10.3103/s1068798x07110020.

[3] R. Clink, Balancing of high-speed four-stroke engines, Proc. Inst. Mech. Eng.: Autom. Divis. 12 (1) (1958) 73–110, http://dx.doi.org/10.1243/pime_auto_1958_000_013_02.

[4] H. Karabulut, Dynamic model of a two-cylinder four-stroke internal combustion engine and vibration treatment, Int. J. Engine Res. 13 (6) (2012) 616–627, http://dx.doi.org/10.1177/1468087412442618.

[5] C. Delprete, C. Gastaldi, L. Giorio, A minimal input engine friction model for power loss prediction, Lubricants 10 (5) (2022) 94, http://dx.doi.org/10.3390/lubricants10050094.

[6] G. Genta, C. Delprete, A. Tonoli, R. Vadori, Conditions for noncircular whirling of nonlinear isotropic rotors, Nonlinear Dynam. 4 (2) (1993) 153–181, http://dx.doi.org/10.1007/bf00045252.

[7] H.Ç. Sezgen, M. Tinkir, Optimization of torsional vibration damper of cranktrain system using a hybrid damping approach, Eng. Sci. Technol., Int. J. 24 (4) (2021) 959–973, http://dx.doi.org/10.1016/j.jestch.2021.02.008.

[8] S.X. Jiang, S.H. Zhang, Z.M. Yu, Study on the field dynamic balance of diesel engine crankshaft, Appl. Mech. Mater. 307 (2013) 299–303, http://dx.doi.org/10.4028/www.scientific.net/amm.307.299.

[9] G. Genta, C. Delprete, E. Brusa, Some considerations on the basic assumptions in rotordynamics, J. Sound Vib. 227 (3) (1999) 611–645, http://dx.doi.org/10.1006/jsvi.1999.2354.

[10] A. Razavykia, C. Delprete, P. Baldissera, Numerical study of power loss and lubrication of connecting rod big-end, Lubricants 7 (6) (2019) 47, http://dx.doi.org/10.3390/lubricants7060047.

[11] K.-H. Suh, Y.-K. Lee, H.-S. Yoon, A study on the balancing of the three-cylinder engine with balance shaft, in: SAE Technical Paper Series, SAE International, 2000, p. 9, http://dx.doi.org/10.4271/2000-01-0601.

[12] J. Saunders, P. Walker, Design and development of a novel balancing system for single and in-line twin cylinder engines, in: SAE Technical Paper Series, SAE International, 2003, pp. 2000–2009, http://dx.doi.org/10.4271/2003-32-0026.

[13] J. Duhovnik, U. Žargi, J. Kušar, M. Starbek, Project-driven concurrent product development, Concurr. Eng. 17 (3) (2009) 225–236, http://dx.doi.org/10.1177/1063293x09343823.

[14] Y. Kang, M.-H. Tseng, S.-M. Wang, C.-P. Chiang, C.-C. Wang, An accuracy improvement for balancing crankshafts, Mech. Mach. Theory 38 (12) (2003) 1449–1467, http://dx.doi.org/10.1016/s0094-114x(03)00097-1.

[15] J.W. Lund, J. Tonnesen, Analysis and experiments on multi-plane balancing of a flexible rotor, J. Eng. Ind. 94 (1) (1972) 233–242, http://dx.doi.org/10.1115/1.3428116.

[16] M.S. Darlow, Balancing of high-speed machinery: Theory, methods and experimental results, Mech. Syst. Signal Process. 1 (1) (1987) 105–134, http://dx.doi.org/10.1016/0888-3270(87)90087-2.

[17] T.P. Goodman, A least-squares method for computing balance corrections, J. Eng. Ind. 86 (3) (1964) 273–277, http://dx.doi.org/10.1115/1.3670532.

[18] S.A. Ahmad, K.P. Mahaveer, Dynamic modeling of automotive engine crankshafts, in: Mechanism and Machine Theory, Elsevier, 1994, pp. 995–1006, http://dx.doi.org/10.1016/0094-114X(94)90067-1.

[19] D.L. Cronin, Shake reduction in an automobile engine by means of crankshaft-mounted pendulums, Mech. Mach. Theory 27 (5) (1992) 517–533, http://dx.doi.org/10.1016/0094-114x(92)90041-f.

[20] L. Cvetićanin, Balancing of flexible rotor with variable mass, Mech. Mach. Theory 16 (5) (1981) 507–516, http://dx.doi.org/10.1016/0094-114x(81)90022-7.

[21] C. Gastaldi, M.M. Gola, Criteria for best performance of pre-optimized solid dampers, in: Volume 7C: Structures and Dynamics, American Society of Mechanical Engineers, 2018, p. 12, http://dx.doi.org/10.1115/gt2018-75961.

[22] E.G.M. Brusa, S. Morsut, Design and structural optimization of the electric arc furnace through a mechatronic-integrated modeling activity, IEEE/ASME Trans. Mechatron. 20 (3) (2015) 1099–1107, http://dx.doi.org/10.1109/tmech.2014.2364392.

[23] C. Gastaldi, T.M. Berruti, M.M. Gola, The relevance of damper pre-optimization and its effectiveness on the forced response of blades, in: Volume 7B: Structures and Dynamics, American Society of Mechanical Engineers, 2017, p. 11, http://dx.doi.org/10.1115/gt2017-64402.

[24] M. Akbari, P. Asadi, M.B. Givi, G. Khodabandehlouie, Artificial neural network and optimization, in: Advances in Friction-Stir Welding and Processing, Elsevier, 2014, pp. 543–599, http://dx.doi.org/10.1533/9780857094551.543.

[25] B.M. Adams, W.J. Bohnhoff, K.R. Dalbey, Dakota, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.15 User's Manual, Sandia National Laboratories, 2021.

[26] Streparava, https://www.streparava.com/prodotti/powertrain/alberi-motore. (Accessed 05 September 2022).

[27] A. Dagna, C. Delprete, C. Gastaldi, A method for crankshaft balancing: theory development and numerical implementation, 2021, Politecnico di Torino.

[28] A. Dagna, C. Delprete, C. Gastaldi, A general framework for crankshaft balancing and counterweight design, Appl. Sci. 11 (19) (2021) 8997, http://dx.doi.org/10.3390/app11198997.

[29] M. Mahrach, G. Miranda, C. León, E. Segredo, Comparison between single and multi-objective evolutionary algorithms to solve the knapsack problem and the travelling salesman problem, Mathematics 8 (11) (2020) 2018, http://dx.doi.org/10.3390/math8112018.

[30] N.A. Zolpakar, S.S. Lodhi, S. Pathak, M. Anand, Sharma, Application of multi-objective genetic algorithm (MOGA) optimization in machining processes, in: Optimization of Manufacturing Processes, 2020, pp. 185–199, http://dx.doi.org/10.1007/978-3-030-19638-7_8.

[31] X. Lin, Z. Yang, X. Zhang, Q. Zhang, Pareto set learning for expensive multi-objective optimization, 2022, http://dx.doi.org/10.48550/ARXIV.2210.08495, arXiv.

[32] L. Resteghini, P.L. Lanzi, R. Nebuloni, C. Riva, C. Capsoni, P. Gabellini, Single-objective genetic algorithm for dynamic optimization of reconfigurable antenna systems, in: 2013 7th European Conference on Antennas and Propagation, EuCAP, 2013, pp. 1333–1335.

[33] D.R. Jones, J.R.R.A. Martins, The DIRECT algorithm: 25 years later, J. Global Optim. 79 (3) (2020) 521–566, http://dx.doi.org/10.1007/s10898-020-00952-6.

[34] R. Hooke, T.A. Jeeves, "Direct search" solution of numerical and statistical problems, J. ACM 8 (2) (1961) 212–229, http://dx.doi.org/10.1145/321062.321069.

[35] E. Gazioğlu, A. Etaner-Uyar, Experimental analysis of a statistical multiploid genetic algorithm for dynamic environments, Eng. Sci. Technol., Int. J. 35 (Turkey) (2022) 101173, http://dx.doi.org/10.1016/j.jestch.2022.101173.

[36] J.S. Arora, Multi-objective optimum design concepts and methods, in: Introduction to Optimum Design, Elsevier, 2012, pp. 657–679, http://dx.doi.org/10.1016/b978-0-12-381375-6.00017-6.