Advances in Engineering Software 208 (2025) 103974

Contents lists available at ScienceDirect

Advances in Engineering Software

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

Research paper

Multi-objective optimisation of complex mechanisms using Moving Spheres: An application to suspension elasto-kinematics

Lorenzo De Santanna, Massimiliano Gobbi *, Riccardo Malacrida, Gianpiero Mastinu

Department of Mechanical Engineering, Politecnico di Milano, Milan 20156, Italy

ARTICLE INFO

Keywords:
Multi-objective optimisation
Feedforward artificial neural networks
Suspension optimal design
Mechanisms

ABSTRACT

This paper presents a new iterative method, called Moving Spheres (MS), for solving multi-objective design optimisation problems involving three-dimensional mechanisms. The method is suited to problems in which most of the design variables belong to the three-dimensional Euclidean space. MS method is able to explore efficiently the design space and identifies the regions where the optimal solutions are located, resulting in a clear spatial representation of optimal solutions. In this paper, MS method is applied to the elasto-kinematic optimisation of an automotive suspension system. The optimal locations of suspension joints are sought within spherical neighbourhoods of a reference suspension. This preserves the kinematic compatibility of the mechanism and facilitates the exploration of the design space through iterative updates of the reference suspension. The rigorous k-optimality metric, which introduces a hierarchical sorting in the Pareto-optimal set, is employed to rank optimal design solutions. In the suspension test case, the Pareto-optimal set of approximated through Moving Spheres method is compared with the Pareto-optimal sets resulting from Parameter Space Investigation and multi-objective optimisation Genetic Algorithm with sorting (KEMOGA) methods, considering similar computational time. Moving Spheres method yields a more accurate approximation of the Pareto-optimal set.

1. Introduction

This work introduces a new multi-objective optimisation method, called Moving Spheres (MS), developed to optimise mechanisms defined in the three-dimensional Euclidean space. The method is specifically designed for problems in which most of the design variables corresponds to the Cartesian coordinates of the joints of the mechanism. The volume in which these mechanism joints are located is referred to as design space. In general, the design space is large, complex and often cluttered. In such scenarios, the exploration of the entire design space at once may be either undesirable or unfeasible.

It is undesirable because ensuring the geometric compatibility required for proper mechanism motion typically requires introducing constraints on the relative locations of the joints. This imposes an aprioristic bias on the exploration process.

It is unfeasible because, without such constraints, the design space exploration results unpractical in terms of computational time, both due to the vast size of the domain and the large number of faulty configurations generated. Moreover, the unconstrained exploration of the design space may lead to mechanisms with shapes that are expensive to be manufactured.

Moving Spheres method addressed these limitations by performing a sequential exploration of the design space, progressively focusing on regions where Pareto-optimal solutions have already been identified. To illustrate the method, this work applies MS to the optimisation of an automotive suspension system. The suspension is optimised for nine objective functions. The number of design variables is twenty-one: five hard points, with three coordinates each, and six scaling factors for the elastic characteristic of the bushings. The constraints imposed on the problem include the available design space for the hard point coordinates and the upper and lower bounds for the variation of the scaling factors of the bushings.

The paper is organised as follows. A literature review about the optimal design of mechanisms with high dimensionality, with a focus on the kinematics of automotive suspension systems, is provided in Section 1.1. In Section 2, the theoretical formulation of multi-objective optimisation problems is presented, together with the metric adopted inside MS method to rank optimal design solutions (k-optimality, Section 2.2). Section 3 describes the suspension problem. The design variables, the constraints and the objective functions are explained. Section 4 introduces Moving Spheres method and the flowchart of the method applied to the suspension case. Results of the case study optimised using Moving Sphere method are discussed and compared with the ones obtained through the Parameter Space Investigation (PSI) method [1,2] and KEMOGA [3] in Section 5.

* Corresponding author.
E-mail address: massimiliano.gobbi@polimi.it (M. Gobbi).

https://doi.org/10.1016/j.advengsoft.2025.103974
Received 10 February 2025; Received in revised form 15 May 2025; Accepted 1 June 2025
Available online 20 June 2025

List of abbreviations	
CD	Cumulative database
DV	Design variable
FNN	Feedforward neural network
HP	Hard point, suspension -
KEMOGA	$k\varepsilon$ Multi-Objective Genetic Algorithm
LC	Load case
MOP	Multi-objective optimisation problem
MS	Moving Spheres
NSGA	Non-dominated Sorting Genetic Algorithm
OWT	Opposite wheel travel
PSI	Parameter Space Investigation
PSO	Particle Swarm Optimisation
PWT	Parallel wheel travel
SF	Scaling factor
TB	Traction and braking

1.1. Literature review

The literature review covers the following topics:

- multi-objective optimisation problems, with a focus on high dimensionality;
- design problems of mechanisms characterised by high dimensionality;
- design of the kinematics of suspension systems.

The first notions of optimality in multi-objective optimisation problems were proposed by Edgeworth [4] and Pareto [5] at the turn of the 20th century. Nearly a hundred years later, the advent of modern computing made possible the development of powerful algorithms, known as Evolutionary Multi-Objective approaches [6]. These class of optimisation algorithms constructs an approximation of the Pareto-optimal set (see Section 2.1) by evolving a population of solutions over successive generations.

In 1994, Srinivas and Deb proposed the Strength Pareto Evolutionary Algorithm (SPEA), which combined Pareto dominance with a fitness assignment mechanism to improve convergence in multi-objective optimisation problems [7]. This approach was further refined in 2000 by Zitzler et al. with SPEA2, a more robust and efficient method for real-world applications [8]. In 2002, Deb et al. introduced Non-dominated Sorting Genetic Algorithm II (NSGA-II), an elitist and computationally efficient genetic algorithm that uses non-dominated sorting and crowding distance to preserve solution diversity while enhancing convergence [9]. Fundamental references about this class of algorithms include the books by Deb [6] and Coello et al. [10].

As the number of objective functions increases, so does the ratio of dominant solutions over dominated [3,11]. When dealing with problems involving a large number of objectives, the Pareto-optimal set may become so extensive that Pareto optimality alone is not sufficient for selecting a final design solution. To address this limitation, Gobbi [3] proposed a hierarchical ranking within the Pareto-optimal set that allows for finer distinction among solutions. A further development of this concept is found in the work by Levi et al. introducing the $k\varepsilon$-optimality [11] on which $k\varepsilon$ Multi-Objective Genetic Algorithm (KEMOGA) is based [3,12].

Referring to design problems of mechanisms characterised by high dimensionality, numerous papers have addressed the design of complex mechanisms or structures. Nevertheless, dealing specifically with issues arising from high-dimensional design spaces remains relatively under-discussed [13]. Gao et al. [14] apply Artificial Intelligence to the design of spatial mechanisms. Zou et al. [15] optimise 3 design variables related to replacing spherical joints with planar joints. Huang et al. [16] focus on a reconfigurable parallel mechanism involving 3 design variables. Zhang et al. investigate the optimisation of a family of manipulators acting on 5 design variables [17] and propose a general procedure for the optimal design of parallel mechanisms [18]. Similar observations apply to publications addressing the optimal design of different types of mechanisms: Ling et al. [19] explore spatial compliant mechanisms using 4 design variables. Guacheta-Alba et al. [20] apply NSGA-II to the design of a mechanism for fused filament additive manufacturing involving 4 design variables. Gu et al. [21] consider 2 design variables in designing over-constrained robotic legs. Shi et al. [22] perform multi-objective optimisation of a cable-strut tensioned antenna using 4 design variables, whereas Pan et al. [23] consider the optimisation of an earthmoving mechanism, reducing the number of design variables from 30 down to 3. None of these papers explicitly addresses how to cope with high dimensionality, despite frequently employing the same tools adopted in the current work.

Concerning extremely high-dimensional design spaces, two relevant works by Zhang et al. [24] and Vo et al. [25] can be mentioned. The former considers 288 design variables in the optimisation of a satellite module layout using a multi-objective cooperative co-evolutionary algorithm. The latter deals with 120 design variables and multi-objective Particle Swarm Optimisation (PSO) algorithms. However, both studies limit complexity predominantly to the design space dimensionality, as number of objective functions considered does not exceed 3.

With regard to the design of suspension systems, the ultimate goal of optimal design is to obtain comfortable [26] and safe [27] vehicles. When addressing the optimal design of automotive suspension kinematics, both scalar and multi-objective optimisation methods are commonly applied [28]. As highlighted by Du et al. [29] in their review on electrically interconnected suspensions, complexity is a common issue in suspension design problems due to the high number of design variables and objective functions involved. For this reason, although they are fundamental to validate simulation outcomes, experimental tests typically involve fewer design variables and are less frequently performed. Notable examples include the works by Gobbi et al. [30,31] on experimental indoor methods aimed at tuning suspension behaviour concerning vibration and harshness, and Derrix et al. [32], which focuses on the influence of chassis stiffness on the kinematic and compliance characteristics of suspensions.

Considering the multi-objective optimisation applied to lumped-parameter vehicle models, Yang et al. provided an analytical solution for a problem involving 3 objective functions and 3 design variables [33], as well as a study employing collaborative optimisation and analytical target cascading approaches involving 6 design variables and 3 objective functions [34]. Concerning elasto-kinematic optimisation of a suspension system using multi-body models, Datoussaid et al. [35] optimised the dynamic behaviour of a vehicle acting on three suspension hard points. Similarly, Simionescu and Beale [36] adopted a single-objective approach to improve kinematic consistency in a five-link rear suspension, considering 30 design variables derived from the coordinates of ten hard points. Choi et al. [37] examined a design problem involving 10 design variables related to suspension stiffness and two objective functions connected to brake and lateral force compliance. Feng et al. [38] considered 6 design variables linked to bushing elastic properties to optimise toe angle and camber angle variations under lateral tyre force. Šagi [39] further explored multi-objective optimisation employing evolutionary algorithms to balance three objectives. Wang et al. [40] applied PSO and NSGA-II algorithms to optimise an electric steering system featuring 10 design variables and 3 objective functions. Similarly, Shi et al. [41] developed a Double Loop PSO algorithm for the robust design optimisation of a MacPherson suspension system, involving 11 design variables (two of which are affected by uncertainty) and 2 objectives functions.

Recent contributions related to the multi-objective optimisation of suspension systems include several noteworthy papers. Bahrami Joo et al. [42] apply a multi-objective genetic programming approach using bond graphs to the multi-objective robust design of an active suspension system. Their work is based on a two-degrees-of-freedom vehicle model, involving 3 design variables and 4 objective functions. Llopis-Albert et al. [43] propose a multidisciplinary framework that accounts for 6 variables and 5 objective functions, including the mass of the suspension arm and two dynamic quantities – the chassis pitch acceleration and the peak equivalent stress on the suspension arm – under two distinct manoeuvres: straight-line braking and braking in turn. Hegazy et al. [44] adopt a multi-objective genetic algorithm to optimise a regenerative MacPherson strut with 3 objective functions, while Lu et al. [45] investigate dynamic toe angle variations in a four-wheel steering vehicle, acting on 8 suspension hard points coordinates.

In contrast to simultaneous approaches, Knapczyk and Maniowski [46] utilise a scalarised method, reducing five objective functions into a single scalar objective. Similarly, Nasiri et al. [47], in optimising a MacPherson suspension by varying the location of 6 joints, adopt a weighted sum of three objectives to define a single objective function. Finally, Arshad and Lodi [48] evaluate the performance of a several algorithms applied to optimise camber angle behaviour in a double wishbone suspension, considering 4 design variables.

Mastinu et al. [49] summarise the theoretical tools required for the optimal design of mechanical systems and introduce the concept of global approximation. In chapter 7 of their book, they optimise the tyre-suspension system of a racing car using Multilayer Perceptron and Radial Basis Neural Networks as global approximators. The same tools have been employed for the optimisation of a road vehicle suspension system featuring 38 design variables and 29 objective functions [50]. Cheng and Lin [51], use Kriging models, whereas Feng et al. [38] adopted Chebyshev polynomials to approximate the response of suspension systems.

2. Multi-objective optimisation

2.1. General formulation of the problem

A multi-objective optimisation problem (MOP), with n design variables and m objective functions is defined as [52]

$$\begin{aligned} &\min_{\boldsymbol{x}\in\mathbb{R}^n} \boldsymbol{f}(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}^m \\ &\text{subject to} \\ &g_i(\boldsymbol{x}) \le 0 \quad i \in [1, q] \\ &\boldsymbol{x}_{lb} \le \boldsymbol{x} \le \boldsymbol{x}_{ub} \end{aligned} \tag{1}$$

where $\boldsymbol{f}(\boldsymbol{x})$ is the vector of m objective functions, $\boldsymbol{x}$ is the vector of n design variables – bounded between a lower and an upper limit ($\boldsymbol{x}_{lb}$ and $\boldsymbol{x}_{ub}$) – and g_i is one of the q constraints. When a vector of design variables $\boldsymbol{x} \in \mathbb{R}^n$ that satisfies the constraints is mapped through the vectorial function $\boldsymbol{f}(\boldsymbol{x})$ into the objective functions space $\mathbb{R}^m$, it becomes a feasible solution of the MOP. A well-posed MOP always has more than one optimal solution. When a feasible solution is compared with other feasible solutions, it can be dominant or dominated [49]. A dominant solution is called Pareto-optimal solution. A solution $\boldsymbol{x}_i$ of the MOP is Pareto-optimal if $\nexists\, \boldsymbol{x}_j$ such that [49]

$$\begin{cases} \quad f_k(\boldsymbol{x}_j) \le f_k(\boldsymbol{x}_i) \quad \forall k \in [1, m] \\ \exists l : f_l(\boldsymbol{x}_j) < f_l(\boldsymbol{x}_i) \end{cases} \tag{2}$$

This statement defines the condition of dominance. A solution is Pareto-optimal if there does not exist another feasible solution that improves at least one objective function without worsening at least another one. The set of Pareto solutions is called Pareto-optimal set or Pareto front, as it forms, in bi- and three-dimensional objective functions spaces, a line or surface that bounds the dominated solutions, facing the origin of the objective functions space.

Pareto optimality check is the first and necessary step to classify solutions in multi-objective problems [49]. As discussed in the Introduction (Section 1), the ratio of dominant to dominated solutions increases with the number of objective functions [5]. This phenomenon is known as the "curse of dimensionality" and the authors of this paper suggest to address it introducing a hierarchical classification inside the Pareto-optimal set.

Table 1
Optimality level of a design solution according to k-optimality, in the context of a multi-objective optimisation problem involving n objective functions. A solution is k-optimal if and only if it is Pareto-optimal for *any* subset of $n-k$ objective functions.

k	Optimality level
0	Minimum (simple Pareto-optimal)
1	⋮
⋮	
$n-1$	Maximum (Utopia)

2.2. Ranking optimal solutions: k-optimality

In this work, the method adopted to hierarchically order the solutions belonging to the Pareto-optimal set is k-optimality [11], where k denotes the optimality level of the solution. Given a MOP with n objective functions, if a solution remains Pareto-optimal for *any* subset of $n-k$ objective functions, it is k-optimal. The optimality level k of a solution varies between 0, for a solution that improves just one objective function with respect to any other solution in the solution set, and $n-1$, for a solution that is not dominated in any of the n objective functions. This latter case is called Utopia, an ideal solution that does not exist in any well-posed MOP, as it would optimise all the objective functions simultaneously. The value k assigned to each solution of the Pareto-optimal set is the maximum number of objective functions that can be arbitrarily removed from the problem definition while still keeping the solution dominant with respect to the others. A higher k means being closer to Utopia. A solution that is simply Pareto-optimal ($k = 0$) might lose its dominance even if a single objective function is removed from the problem. A solution with a higher k is not only closer to Utopia but is also guaranteed to remain Pareto-optimal for any combination of $n-k$ objective functions arbitrarily taken from the problem definition. Table 1 summarises the ranking of Pareto-optimal solution according to k-optimality.

2.2.1. Optimality level computation

This section explains the algorithm for the computation of the k-optimality. It is assumed to have p design solutions, each described by m objective functions f_k. Consider a step function $S(x) : \mathbb{R} \to \mathbb{R}$ such that

$$S(x) = \begin{cases} 1 & \text{if } x \le 0 \\ 0 & \text{if } x > 0 \end{cases} \tag{3}$$

Then, the k-optimality algorithm is

1. Take the design solution i;
2. Take another design solution j such that $i \ne j$;
3. $c_{j,i} = \sum_{k=1}^{m} S\left(f_k(\boldsymbol{x}_i) - f_k(\boldsymbol{x}_j)\right)$;
4. Repeat points 2 and 3 until all the $p-1$ design solutions are considered;
5. $k_{opt,i} = \min(c_{j,i})$.

where $k_{opt,i}$ is the optimality level of the i-th design solution.

3. Case study: suspension design problem

3.1. Design variables

The MacPherson suspension studied in this paper, shown in Fig. 1, is optimised by acting on twenty-one design variables, grouped into the two sets shown in Table 2. The first set is composed by the spatial variables: the three coordinates of each hard point (HP) that defines the

Table 2
The suspension test case includes 21 design variables (DVs) grouped into two sets: Spatial and Elastic. Spatial DVs are the Cartesian coordinates of the hard points, while Elastic DVs represents the scaling factors of the bushings.

Design variables (DVs)	
Dimension of the design space	
21	
Spatial DVs	Elastic DVs
15	6

Table 3
Number (HP) and name (HP Name) of the five hard points defining the MacPherson suspension. Coordinates values are expressed in mm. The HP number corresponds to the identifiers used in Fig. 1.

HP	HP Name	x	y	z
1	lca_front	60	−400	190
2	lca_outer	240	−700	175
3	lca_rear	460	−390	205
4	tierod_inner	467	−400	330
5	tierod_outer	410	−690	300

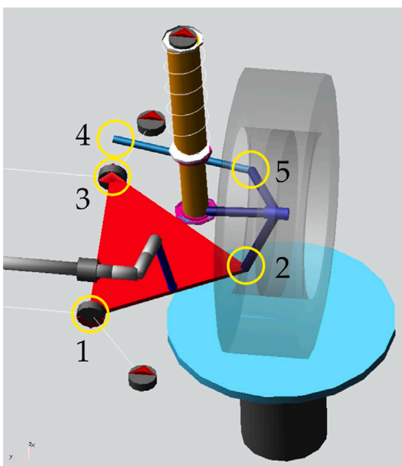

Fig. 1. Reference configuration of the MacPherson suspension used in the test case. The five hard points listed in Table 3, along with two bushings whose scaling factors are defined in Eq. (5), constitute the design variables of the problem.

MacPherson suspension. The second set consists of the scaling factors of the elastic force curves of the bushings.

Spatial DV. The hard points that defines the MacPherson suspension are five. There are fifteen spatial variables in total, corresponding to the three coordinates (x, y and z) for each hard point. The number (HP), the name (HP Name) and the coordinates of the hard points are collected in Table 3.

Elastic DV. Bushing forces are described by three tabular curves (force versus displacement), one for each principal direction. A scaling factor (SF) is a scalar multiplier applied to the curves to decrease or increase the force produced by the bushing at a given displacement (see Fig. 2). Scaling factors provide a simple and effective way to manage the force characteristics of bushings [53].

The force F along a principal direction $i = x, y, z$ of the bushing mounted on the hard point HP is computed using Eq. (4). Possible trends of the force, together with the bushing principal coordinates, are shown in Fig. 2. The nominal force F_0 is obtained by multiplying the displacement ξ_i with the non-linear stiffness $k(\xi_i)$; this nominal force is then adjusted by the scaling factor $\mathrm{SF}_{\mathrm{HP},i}$

$$F\left(\mathrm{SF}_{\mathrm{HP},i}, \xi_i\right) = \mathrm{SF}_{\mathrm{HP},i} \cdot F_0\left(\xi_i\right) = \mathrm{SF}_{\mathrm{HP},i} \cdot k\left(\xi_i\right) \cdot \xi_i \tag{4}$$

The scaling factors treated as design variables refer to the bushings mounted on HPs 1 and 3 in Fig. 1. These bushings absorb most of

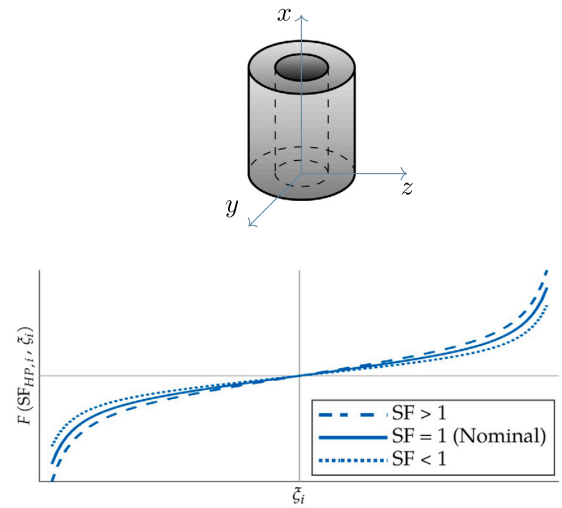

Fig. 2. Bushing principal directions (top) and effect of the scaling factor on the force characteristic F of the bushing (bottom).

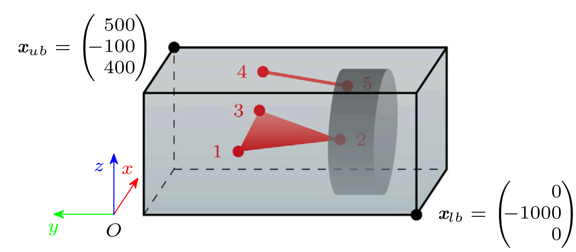

Fig. 3. Three-dimensional design space for the hard points of the MacPherson suspension shown in Fig. 1. The grey cylinder represents the wheel (not included in the optimisation). The lower bound $\boldsymbol{x}_{lb}$ and the upper bound $\boldsymbol{x}_{ub}$ of the design space are expressed in millimetres.

the loads coming from the road and thus, through their deformation, significantly influence the elastic response of the system. As a result, the Elastic DVs of the design problem can be expressed as

$$\boldsymbol{x}_e = \left[\mathrm{SF}_{1,x} \;\; \mathrm{SF}_{1,y} \;\; \mathrm{SF}_{1,z} \;\; \mathrm{SF}_{3,x} \;\; \mathrm{SF}_{3,y} \;\; \mathrm{SF}_{3,z}\right] \tag{5}$$

The nominal value of each SF is equal to 1.

3.2. Constraints

The design variables of the problem are subject to design constraints.

Spatial DV. The hard points must remain within the design space. Fig. 3 represents the design space. The design space used by MS method can take any shape, in this work a rectangular box-shaped volume is adopted for clarity.

Elastic DV. The scaling factors of the force curves of bushings vary in the interval [0.8, 1.2], so that the elastic force at any bushing displacement deviates at most ±20% from the nominal value.

As discussed in the Introduction (Section 1), a suspension system must satisfy additional kinematic constraints in order to operate correctly. These constraints can be explicitly imposed by writing a set of relations that must be satisfied by the coordinates of the hard points. For example, it might be imposed that the x-coordinate of HP1 remains lower than that of HP3 (see Fig. 1). This approach implements aprioristic design restrictions which might result in a biased exploration of the design space.

Another way to address kinematic constraints is to allow the HP locations vary within spherical neighbourhoods of a reference suspension configuration. If the reference suspension is feasible, perturbations

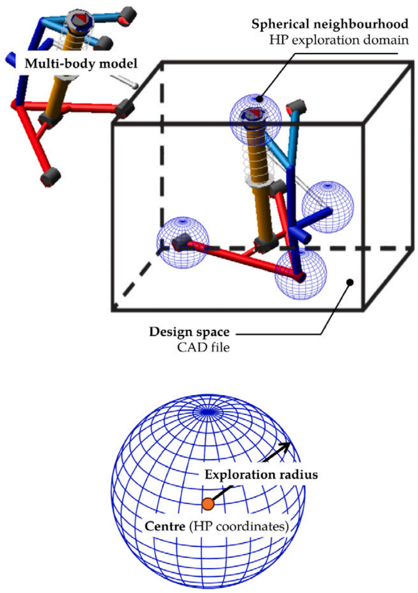

Fig. 4. Representation of the exploration domain of Moving Spheres method (top) and geometrical description of the spherical neighbourhood (bottom) in terms of its radius and the coordinates of its centre. Perturbed suspension configurations are selected within the spherical neighbourhoods of a reference suspension configuration. This set of neighbourhoods can be updated to explore the design space.

of its HPs are unlikely to lead to unfeasible configurations. These perturbed suspensions provide different responses, as the kinematic behaviour of three-dimensional mechanisms is highly non-linear.

This second approach, illustrated in Fig. 4, avoids the need to enforce explicit compatibility relationships. The design space is explored by updating the reference suspension configuration. After each update, the location of neighbourhoods is modified, allowing the generation of new perturbed suspension configurations. The admissible volume for generating perturbed suspension configurations is defined as the intersection between the neighbourhoods and the design space. As described in detail in Section 4.1, this neighbourhood-based exploration of the available design space constitutes the core of the method presented in this work. The scaling factors, by contrast, always vary within their lower and upper bounds.

3.3. Objective functions

The suspension model is simulated using ADAMS Car and the VI Automotive plug-in, within a quasi-static multi-body framework. The analysis involves applying an input to the wheel and evaluating the output of the model in terms of suspension characteristic curves. The input can be either a force or a displacement (see Table 4), and the characteristic curves represent either geometrical or physical quantities. Each analysis reproduces a load case (LC) that reflects a real driving scenario. The three load cases considered in this work are summarised in Table 4. PWT and OWT are mainly kinematic, as the suspension is moved by imposing a vertical displacement on the wheels: symmetric in PWT, which simulates heave motion of the car body; anti-symmetric in OWT, which simulates roll motion. TB, on the other hand, accounts for the compliance of the suspension, as the input is a longitudinal force applied at the tyre contact patch, in both the positive (tractive) and negative (braking) directions.

Table 5 lists all the suspension characteristic curves for each LC described in Table 4. For the toe angle variation in PWT – commonly referred to as "bump steer" – the curvature of the characteristic curve with respect to wheel travel is considered among the objective functions, since a linear trend during compression is generally preferred by automotive designers [54].

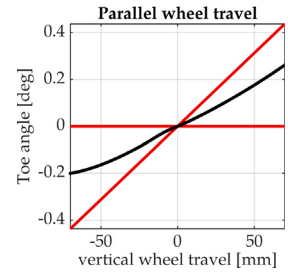

Fig. 5. Toe angle as a function of vertical wheel travel (black line) for the PWT load case (see Table 4). Acceptance bounds, shown in red, define the region of acceptable performance for the considered output.

Table 4
Load cases (LCs), abbreviated as PWT, OWT, and TB, used to describe specific inputs applied to the suspension shown in Fig. 1.

LC	Load case name	Load case input
PWT	Parallel wheel travel	Vertical wheel travel
OWT	Opposite wheel travel	Vertical wheel travel
TB	Traction and braking	Force at contact patch

Table 5
Outputs considered for each of the load cases presented in Table 4. •: Included; ∘: Not included.

Load case output	PWT	OWT	TB
Toe angle	•	•	•
Curvature of toe angle	•	∘	∘
Camber angle	•	•	•
Wheel travel track	•	∘	∘
Wheel travel base	∘	∘	•

Fig. 5 shows a typical output of a multi-body suspension simulation. The black line represents the variation of the toe angle with respect to vertical wheel travel, while the red lines define the acceptance bounds. Each characteristic curve is assigned two acceptance bounds, which are used to compute the corresponding objective function. When the trend of the characteristic curve lies within the acceptance bounds, it is considered acceptable and the objective function takes on a low value; when it falls outside, it is considered unsatisfactory and results in a higher value of the objective function.

As the simulations are quasi-static, output curves are discretised into a prescribed number of points, referred to as increments. As a first step a value $v \in [0, 10)$ is assigned to each increment of the characteristic curve. This value quantifies how much the curve deviates from the most desirable trend, assumed to be half-way between the two acceptance bounds. Therefore, when the increment lies exactly in the middle of the bounds, $v = 0$, while $v \to 10$ as the distance from the bounds tends to infinity (in either direction). If a point of the curve coincides with one of the two bounds, then $v = 1$. The value of v is computed by Eq. (6)

$$\begin{cases} v(x_i) = \left(\dfrac{f(x_i) - t(x_i)}{b_1(x_i) - t(x_i)} \right)^2 & \text{if} \quad f(x_i) \in \left[b_1(x_i),\, b_2(x_i) \right] \\ v(x_i) = 10 - 9\, e^{-\frac{|f(x_i) - b_\star(x_i)|}{\lambda}} & \text{if} \quad f(x_i) \notin \left[b_1(x_i),\, b_2(x_i) \right] \end{cases} \tag{6}$$

where $f(x_i)$ is value of the output, x_i is the value of the input, $b_1(x)$ and $b_2(x)$, with $b_1(x) < b_2(x) \forall x$, are the acceptance bounds. The target value $t(x_i)$ is defined as

$$t(x_i) := \frac{b_1(x_i) + b_2(x_i)}{2}$$

When output is within the interval $[b_1(x_i), b_2(x_i)]$, $v(x_i)$ is computed as a squared normalised residual. If the curve lies outside the acceptance bounds, the value of v grows according to an exponential law.

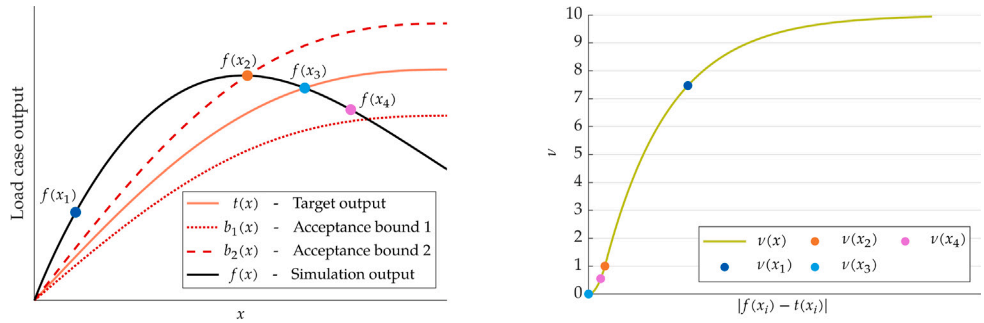

Fig. 6. The chart on the left shows a generic simulation output curve $f(x)$, the target $t(x)$ and the acceptance bounds associated with it. The load of the simulation is indicated by x. The chart on the right shows the trend of ν, computed through Eq. (6). Values $\nu(x_i)$, corresponding to different simulation outputs $f(x_i)$, are highlighted.

Specifically, $b_\star(x_i)$ denotes the closest bound to $f(x_i)$, defined as

$$b_\star(x_i) := \underset{j=1,2}{\arg\min} \left| f(x_i) - b_j(x_i) \right|$$

while λ is a real positive number, defined as $\lambda := 2 \cdot \max \left| b_1(x_i) - b_2(x_i) \right|$. The trend of ν with respect to distance from the target is reported in Fig. 6.

After a value of ν is assigned to all increments, the objective function associated with the given characteristic quantity – here identified as f_{OB} – is computed as the arithmetic mean of all ν values (7)

$$f_{OB} = \frac{1}{N} \sum_{i=1}^{N} \nu(x_i) \tag{7}$$

where N represents the number of increments in the simulation.

3.4. Suspension case study MOP formulation

According to the notation of Eq. (1), the multi-objective optimisation problem considered in this paper can be formulated as

$$\begin{aligned}
&\min_{\boldsymbol{x}} \ \boldsymbol{f}(\boldsymbol{x}) : \mathbb{R}^{21} \rightarrow \mathbb{R}^{9} \\
&\boldsymbol{x} = \left[\boldsymbol{x}_{\mathrm{S}},\ \boldsymbol{x}_{\mathrm{E}}\right]^T \\
&\text{where} \\
&\boldsymbol{x}_{\mathrm{S}} \in \mathbb{R}^{15}; \quad \boldsymbol{x}_{\mathrm{E}} \in \mathbb{R}^{6} \\
&\text{subject to} \\
&g_{\mathrm{HP}}(\boldsymbol{x}_{\mathrm{S}}) \leq 0 \quad \mathrm{HP} \in [1,\ 5] \\
&\boldsymbol{x}_{lb} \leq \boldsymbol{x} \leq \boldsymbol{x}_{ub}
\end{aligned} \tag{8}$$

where $\boldsymbol{x}$ represents the vector of the design variables. The lower and upper bounds for the design variables are collected in the vectors $\boldsymbol{x}_{lb}$ and $\boldsymbol{x}_{ub}$, respectively. The inequality constraints g_{HP} apply only to the Spatial design variables ($\boldsymbol{x}_{\mathrm{S}}$), ensuring that these remain within the spherical neighbourhoods of the hard points (see Section 3.2). This constraint can be expressed as

$$g_{\mathrm{HP}} = (x_{\mathrm{HP}} - x_{c,\mathrm{HP}})^2 + (y_{\mathrm{HP}} - y_{c,\mathrm{HP}})^2 + (z_{\mathrm{HP}} - z_{c,\mathrm{HP}})^2 - R_{\mathrm{HP}}^2 \tag{9}$$

where HP denotes the hard point number (see Table 3). Here, x_{HP}, y_{HP}, z_{HP} are the design variables, while $x_{c,\mathrm{HP}}$, $y_{c,\mathrm{HP}}$, $z_{c,\mathrm{HP}}$ and R_{HP} represent the centre and the radius of the spherical neighbourhood, respectively. As highlighted in Section 3.2, the values of $\boldsymbol{x}_{\mathrm{E}}$ are bounded within the interval $[0.8,\ 1.2]$, while the bounds of $\boldsymbol{x}_{\mathrm{S}}$ are presented in Fig. 3. The design variables and the set of objective functions are summarised in Tables 6 and 7, respectively.

Optimal solutions of a MOP form the Pareto-optimal set, i.e. the set of all the solutions that outperform any other solution in at least one objective function (see Section 2.1). As detailed in Section 2.2, this work adopts the k-optimality metric to refine the selection of optimal solutions within the Pareto-optimal set. From an operational standpoint, since both the Pareto-optimal set and the optimality level k are determined through a direct sorting of the solution set, the algorithm described in Section 2.2.1 allows the simultaneous identification of Pareto-optimal solutions and their corresponding k-optimality levels.

Table 6
Summary of the design variables of the optimisation problem. Suspension hard points are shown in Fig. 1 and listed in Table 3.

Variable	Details
$\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix}$	Coordinates of hard point 1
$\begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix}$	Coordinates of hard point 2
$\begin{pmatrix} x_3 \\ y_3 \\ z_3 \end{pmatrix}$	Coordinates of hard point 3
$\begin{pmatrix} x_4 \\ y_4 \\ z_4 \end{pmatrix}$	Coordinates of hard point 4
$\begin{pmatrix} x_5 \\ y_5 \\ z_5 \end{pmatrix}$	Coordinates of hard point 5
$\begin{pmatrix} \mathrm{SF}_{1,x} \\ \mathrm{SF}_{1,y} \\ \mathrm{SF}_{1,z} \end{pmatrix}$	Scaling factors of the bushing mounted on hard point 1
$\begin{pmatrix} \mathrm{SF}_{3,x} \\ \mathrm{SF}_{3,y} \\ \mathrm{SF}_{3,z} \end{pmatrix}$	Scaling factors of the bushing mounted on hard point 3

Table 7
Summary of the load cases (LCs) and corresponding simulation outputs used to define the set of objective functions $\boldsymbol{f}$. The method adopted for computing the objective functions, along with the associated load cases, is detailed in Section 3.3.

Function	LC	Simulation output
f_1	PWT	Toe angle variation
f_2	PWT	Camber angle variation
f_3	PWT	Wheel track variation
f_4	OWT	Toe angle variation
f_5	OWT	Camber angle variation
f_6	TB	Toe angle variation
f_7	TB	Camber angle variation
f_8	TB	Wheel base variation
f_9	PWT	Curvature of the trend of toe angle variation

4. Multi-objective optimisation of a MacPherson suspension using moving spheres

This Section introduces Moving Spheres (MS) method through an industry-relevant test case. The workflow of the optimisation process is illustrated in Fig. 7, while the multi-objective problem under investigation has been described in Section 3.

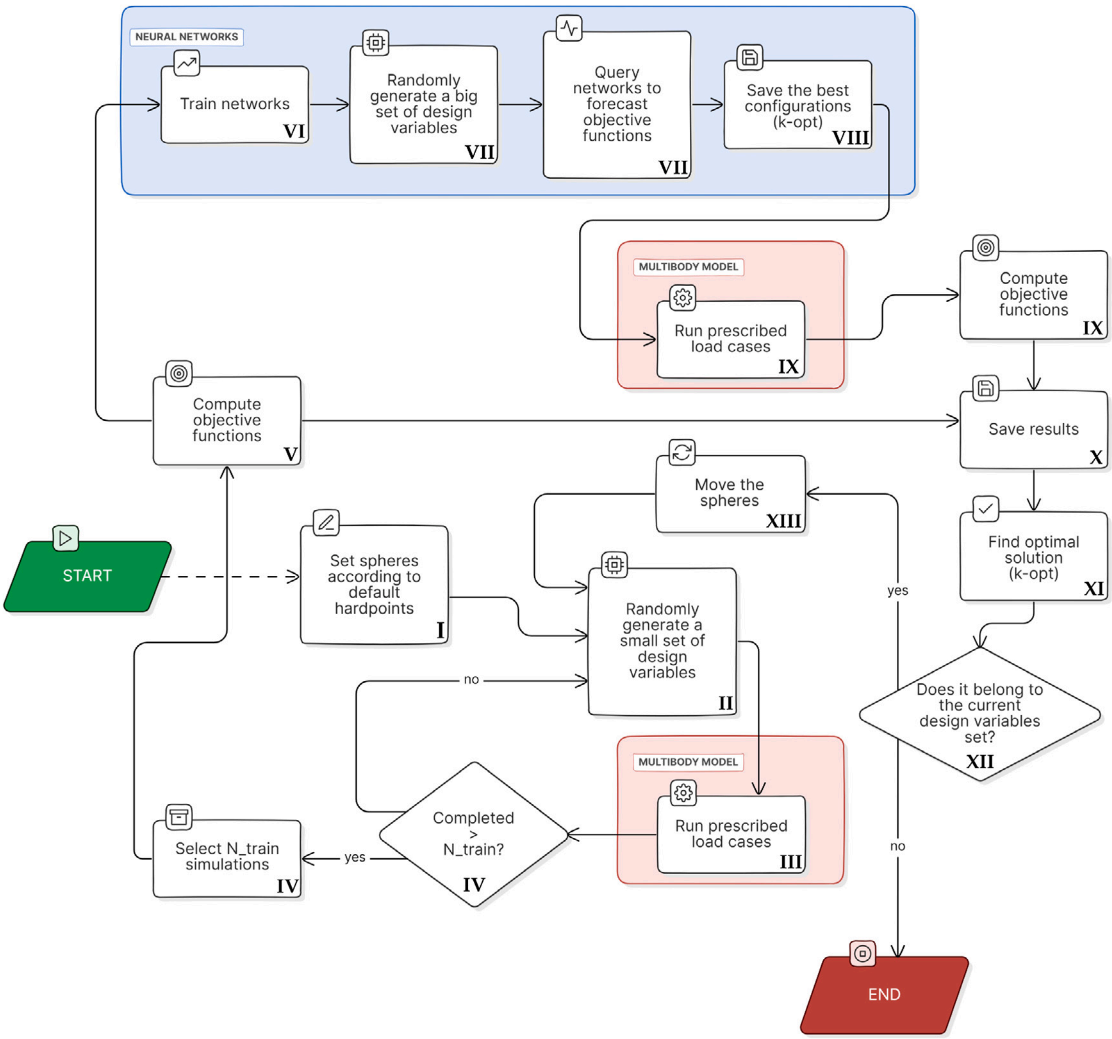

Fig. 7. Workflow of Moving Spheres method. Roman numbers refer to the list of Section 4.1.1, where the procedure is described in detail.

4.1. Moving spheres method

Moving Spheres method is an iterative optimisation method. The flowchart shown in Fig. 7 outlines the procedure. Each iteration begins from a reference suspension configuration, in the neighbourhood of which other configurations are defined and selected. These configurations are then evaluated against a prescribed set of load cases using the objective functions described in Section 3.3. Data on objective functions and design variables are used to train feedforward neural networks (FNNs), which serve to approximate the suspension's multi-body model response. Each FNN receives the full set of the design variables as input and returns one objective function as output; hence, the number of FNNs trained corresponds to the number of objective functions. These networks are then employed to evaluate the performance of millions of candidate configurations. The predicted performance is used to compute the k-optimality level (see Section 2.2) associated with each suspension configuration. The best-performing configurations in terms of k-optimality are validated using the multi-body software, meaning that the objective functions are properly calculated using the original model rather than the neural networks.

At each iteration, the best validated suspension configuration is selected as the new reference. Spherical neighbourhoods are then re-centred on the updated reference joint positions. As a result, the centres of the spherical neighbourhoods shift from the hard points of the previous iteration's reference suspension to the newly selected one, hence the name Moving Spheres. This approach results in a greedy exploration of the design space, as each spherical neighbourhood moves in the most promising direction within the portion of the design space already considered. Fig. 8 illustrates this stepwise exploration for hard point 2 (see Fig. 1). As discussed in Section 3.2, this exploration strategy concerns only the Spatial DVs. The Elastic DVs always vary within fixed bounds throughout the process. The iterative procedure terminates when the stopping criterion described in Section 4.1.1 is satisfied.

4.1.1. In-depth description of the method

Referring to Fig. 7, the steps of the optimisation procedure are described:

I **Set spheres according to initial hard point locations**
The hard point locations of the initial suspension configuration are the centres the of spherical neighbourhoods (see Fig. 4). These neighbourhoods, referred to as spheres, have a user-defined radius. A fixed value of 20 mm was used in this work. The intersection between the spheres and the design space (see Fig. 3) defines the exploration domain for selecting the hard points.

II **Randomly generate a set of suspension configurations**
A batch of suspension configurations (2000 in this case study) is generated within the spheres using uniform stochastic distributions. The scaling factors are also sampled using uniform distribution, constrained within their bounded intervals (see Section 3.2). This step samples the neighbourhood of the reference suspension configuration.

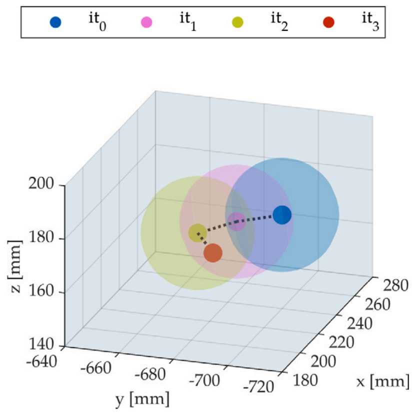

Fig. 8. Shift in the neighbourhood of hard point 2 (see Fig. 1) during the first three iterations of Moving Spheres method. At each iteration, new designs are selected within the spherical neighbourhoods of the hard points, and the best suspension configuration is selected. The centres of the neighbourhoods are then updated to match the hard points of the best configuration. The centres at iterations 0, 1, 2 and 3 are shown.

Table 8
Parameter settings of the genetic algorithm used to optimise the architecture of the surrogate model and the neural network parameters.

Genetic algorithm parameters	
Population size for the genetic algorithm	30
Maximum number of generations	8
Number of stall generations	3
Genetic algorithm variables range	
Number of neurons in the first hidden layer	[10, 20]
Number of neurons in the second hidden layer	[10, 15]
Seed for the random number generator	[0, 20]
Neural network parameters	
Activation function of the hidden layers	Logistic sigmoid
Activation function of the output layer	Linear
Training function used	Bayesian regularisation [55]
Number of epochs	1000
Initial learning rate	$1{\cdot}10^{-3}$
Rate of learning rate decrease	$1{\cdot}10^{-1}$
Rate of learning rate increase	10
Gradient threshold for stopping the training	$1{\cdot}10^{-7}$
Error increase threshold	6
Fraction of data used for training	85%
Fraction of data used for validation	15%
Method for data splitting	Random

III **Run prescribed load cases**
The generated suspension configurations are processed with ADAMS. Each simulation consists of three load cases which apply different loads to the suspension (see Table 4). The outputs for each load case (see Table 5) are then stored.

IV **Check the amount of successful simulations**
Some generated suspension configurations may not be successful. A suspension configuration is considered successful only if all the prescribed load cases can be completed. The number of successful configurations N might be insufficient to train the artificial neural networks. Defining a general threshold N_{train} is not straightforward; it mainly depends on the number of design variables and objective functions. In this work $N_{train} = 1500$. If $N < N_{train}$, steps II and III must be repeated. To ensure consistent training data, if $N \geq N_{train}$, the excess simulations are discarded.

V **Compute objective functions**
The objective functions of the N_{train} suspension configurations are computed as described in Section 3.3.

VI **FNN Training**
Feedforward neural networks with two hidden layers and one output layer are built and trained. Each objective function is approximated by a separate neural network; thus, the number of trained FNNs equals the number of objective functions. The architecture of the FNNs – including the number of neurons in the hidden layers and the seed for the random number generator (used to initialise the network parameters and divide training and validation sets) – is optimised through a genetic algorithm. The fitness FIT_{FNN} of each individual is evaluated as

$$FIT_{FNN} = 1000 \cdot \sigma^2 - R^2 + 1 \tag{10}$$

where σ^2 is the population variance and R^2 is the coefficient of determination, both computed with respect to the entire N_{train} dataset. Table 8 summarises the FNN training parameters.

VII **Query networks with suspension configurations**
The trained artificial neural networks are used to process millions of suspension configuration, all sampled within the design domains defined by the HP and SF ranges used in generating the training data.

VIII **Retain the best performance predictions**
The optimality level of each configuration is assessed using the k-optimality metric [11], as described in Section 2.2. The 500 best suspension configurations according to this metric are retained.

IX **Validate the suspensions found by the FNNs**
The retained configurations are simulated in ADAMS to validate the neural network predictions and retrieve the actual trends of the characteristic curves. FNNs estimate only the objective function values and not the full behaviour of the suspension; hence, validation through multi-body simulation is required.

X **Cumulative database for validated configurations**
Once multi-body simulations of the retained suspension configurations are completed, these validated solutions – along with their objective function values – are stored in a cumulative database (CD). This database includes both the original training simulations (N_{train}) and the additional validated results from FNN predictions. No artificial neural network forecasts flow into the cumulative database. The database contains only validated design solutions. This database cumulates data over iterations of MS method.

XI **Compute the k-optimality and find the optimal solution**
The k-optimality is evaluated among the cumulative database. Although only one suspension configuration must be chosen to start a new iteration (see Fig. 8), multiple configurations may share the highest k-optimality level. In such cases an additional selection criterion must be applied. In this work, the solution with the lowest average of the objective functions is chosen.

XII **Check convergence of the method**
Based on the identified optimal solution, three scenarios are possible:

- If the best suspension belongs to a previous iteration, further exploration is unnecessary and the algorithm should stop.
- If the best suspension is among the N_{train} configurations used for FNN training in the current iteration, it means the FNNs did not identify improvements in the current region; the algorithm should also stop.
- Otherwise, the latest exploration has yielded better solutions, and a new iteration is initiated.

The method may be stopped at any iteration. It is also possible to halt domain exploration and repeat only steps VII to XI, enabling deeper sampling of the current region.

XIII **Move the spheres**
As highlighted in step XI, selecting the optimal suspension configuration is necessary to start a new iteration. As shown in Fig. 8, if another iteration is to be performed, the centres of the spheres are relocated to the hard points of the optimal configuration identified. The process then resumes from step II.

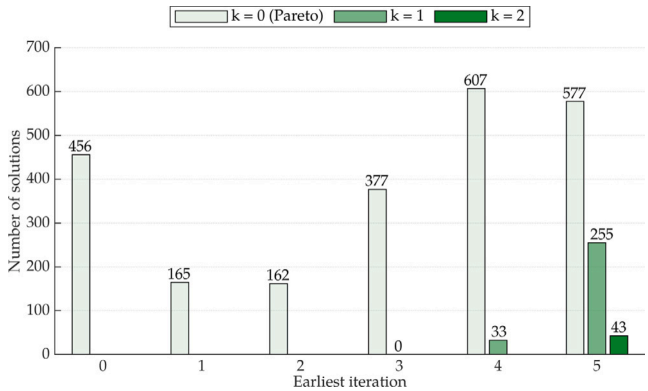

Fig. 9. Optimality level k of the Pareto-optimal design solutions stored in the CD+ database. A total of six iterations were performed. Since the spherical neighbourhoods partially overlap when shifted from one iteration to the next (see Fig. 8), some optimal solutions found in later iterations may lie in regions already explored in earlier ones. For each solution, the earliest iteration E at which it could have been discovered is highlighted. This ensures that all iterations up to E are necessary, as the solution could not have been found in any of the previous $E-1$ iterations.

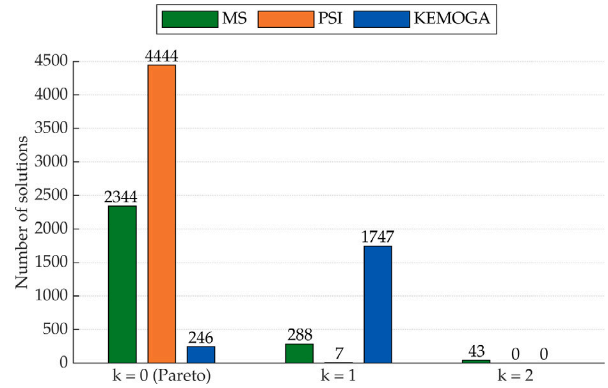

Fig. 10. Pareto-optimal solutions from the CD+ database, grouped by k-optimality level. MS method yields configurations with higher optimality level.

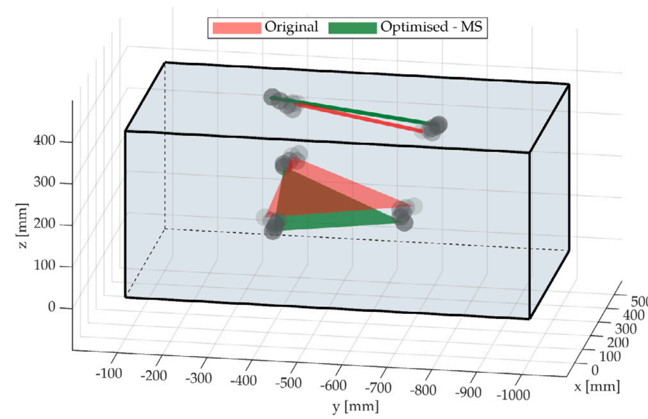

Fig. 11. Representation of one 2-optimal configuration selected to represent the MS method (see Fig. 12), shown in comparison with the reference case. The lower control arm and tie rod, which are the structural components subject to optimisation, are highlighted. Spheres, becoming progressively darker, illustrate the evolution of the spherical neighbourhoods throughout the iterative procedure.

5. Results and comparative analysis

This section illustrates the results of the optimisation of the MacPherson suspension system presented in Section 3. The performance of Moving Spheres method is first examined in Section 5.1, while Section 5.2 presents a comparison with the Parameter Space Investigation method [1,2] and KEMOGA [10].

5.1. Moving spheres performance

A total of six iterations of MS method were performed, leading to a gross amount of $1.2 \cdot 10^9$ suspension configurations processed by the FNNs and 12 000 validated design solutions stored in the cumulative database (CD). These validated solutions are used to construct the Pareto-optimal set. To demonstrate the effectiveness of Moving Spheres method, it must be shown that the progressive shifting of the spheres leads to solutions that dominate those found in earlier iterations. If this is the case, the highest-quality configurations in terms of optimality level will mostly be found in the final iteration.

Fig. 9 shows the distribution of the Pareto-optimal set, constructed using the CD, across the six iterations of MS method. Dominant solutions are grouped into bars according to their optimality level: 0-optimal or Pareto ($k = 0$), 1-optimal ($k = 1$) and 2-optimal ($k = 2$). The spheres at a given iteration partially overlap with those from the previous iteration. If an optimal solution is found during iteration E but lies within regions already explored in iteration $E-1$, then the additional computational effort of iteration E was unnecessary (see step XII in Section 4.1.1). Fig. 9 shows that most of the 1-optimal and 2-optimal solutions in the CD were discovered at iteration 5, and that these solutions could not have been found in any of the earlier iterations. This outcome confirms that moving the spheres leads to an improvement in solution quality according to the k-optimality metric. Similar conclusions can be drawn from Fig. 13, which shows the hard point locations of all Pareto-optimal solutions identified using Moving Spheres method.

5.2. Comparative analysis

The approximation of the Pareto-optimal set obtained using Moving Spheres method is compared with a traditional Parameter Space Investigation method and an application of KEMOGA.

The PSI method relies on uniform sampling of the design space [1, 2]. The number of PSI suspension configurations to be evaluated, 25 000, was selected so that the computational time required for objective function evaluation matches the total time spent by MS method (including multi-body simulations, FNNs training, and FNNs querying over six iterations). Therefore, the PSI method implemented here does not exploit the surrogate model of MS method.

Table 9
Coordinates of the five hard points (HPs) of the optimal suspension configuration obtained at the end of Moving Spheres iterations (iteration 5 in Fig. 9). The Euclidean distances of the five optimised HP locations relative to their original positions (iteration 0 in Fig. 9) are also reported. The original HP coordinates are listed in Table 3 and illustrated in Fig. 1. Values are expressed in millimetres.

HP	x	y	z	Distance
1	1.0	−417.7	163.5	67.0
2	205.9	−680.1	127.2	62.0
3	458.5	−366.5	162.6	48.5
4	476.5	−340.0	333.2	60.8
5	450.1	−730.7	295.7	57.3

Table 10
Scaling factors (SFs) of the optimal suspension configuration at the end of Moving Spheres iterations (iteration 5 in Fig. 9). Each SF has a nominal value equal to 1 (see Section 3.1).

HP	Bushing name	$\mathrm{SF}_{\mathrm{HP},x}$	$\mathrm{SF}_{\mathrm{HP},y}$	$\mathrm{SF}_{\mathrm{HP},z}$
1	sf_lca_front	1.10	1.11	1.15
3	sf_lca_rear	0.85	0.95	1.19

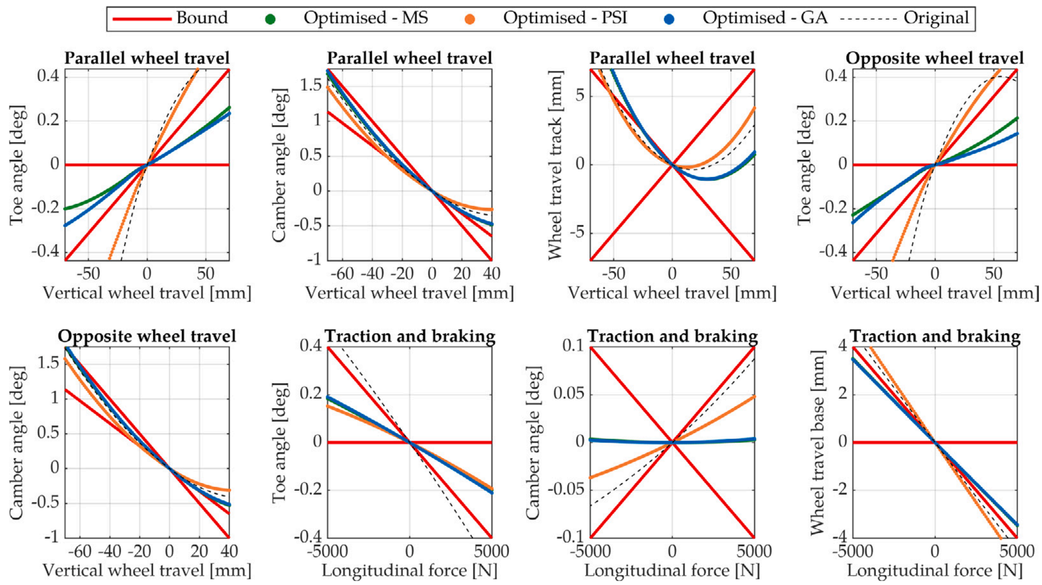

Fig. 12. Characteristic curves (see Table 5) for the optimal suspension configuration obtained at the end of MS iterations (see Fig. 11), and for the suspensions optimised through the PSI and KEMOGA methods (one configuration selected among the 1-optimal solutions for each, see Fig. 10), shown in comparison with the characteristic curves of the original suspension. Acceptance bounds are shown in red.

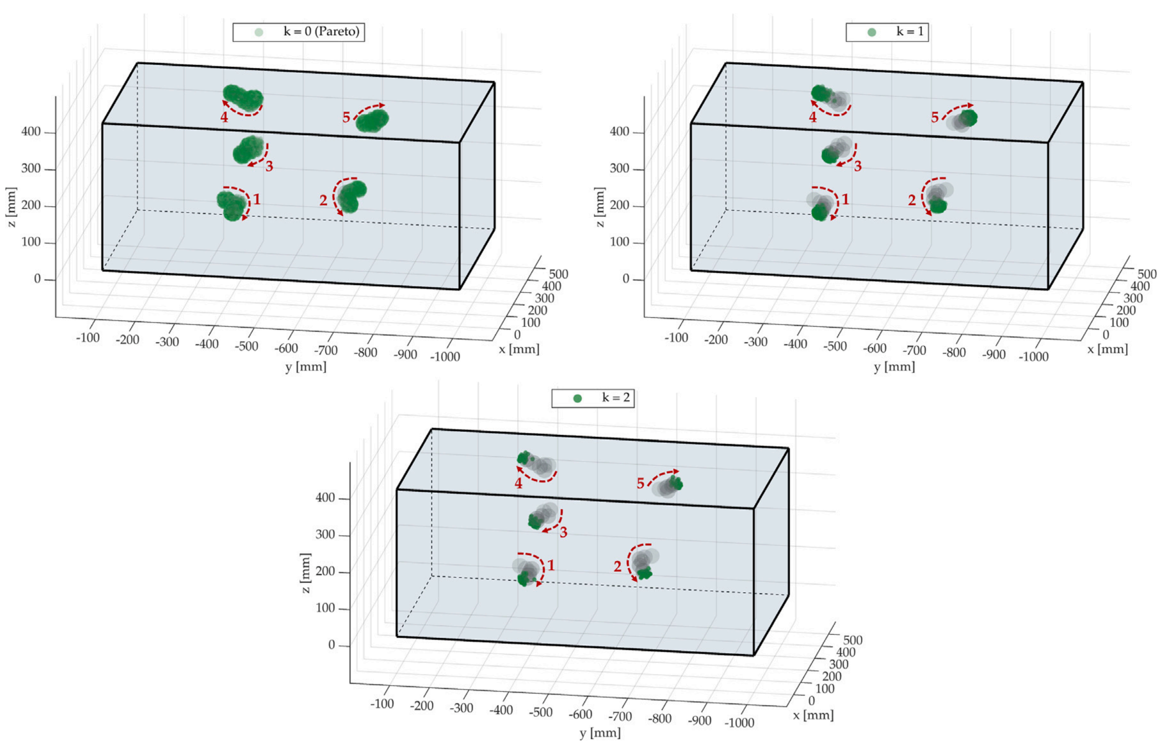

Fig. 13. Representation of the hard points of all Pareto-optimal solutions obtained using Moving Spheres method, grouped by their k-optimality level. The numbers corresponds to the hard point ordering shown in Fig. 1. The arrows qualitatively indicate the movement of the spheres during the iterations of Moving Spheres method.

Suspension configurations are selected within the envelope of the spheres defined by MS method to ensure a reasonable number of successful suspension configurations (see step IV in Section 4.1.1). To achieve this, the HPs of candidate configurations are initially sampled within the rectangular box-shaped volume tangent to the envelope of spheres, while the scaling factors are drawn within the prescribed range. Configuration generation employs a Mersenne Twister pseudorandom number generator [56]. Configurations falling outside the volume enclosed by the spheres are discarded. This process is repeated until the required number of configurations is obtained.

The $k\epsilon$ Multi-Objective Genetic Algorithm (KEMOGA, [10]) implemented in this study is based on the concept of k-optimality. Each design variable is encoded into a binary string (gene), so that the range of each variable – mirroring that used for PSI – is divided into $2^{n_{bit}}$ equally spaced levels, where n_{bit} denotes the gene length. Suspension configurations are represented by a concatenation of genes forming a chromosome, each of which defines an individual in the genetic algorithm. At each generation, the performance of each individual is estimated using the surrogate model from MS method, and fitness is assigned as follows:

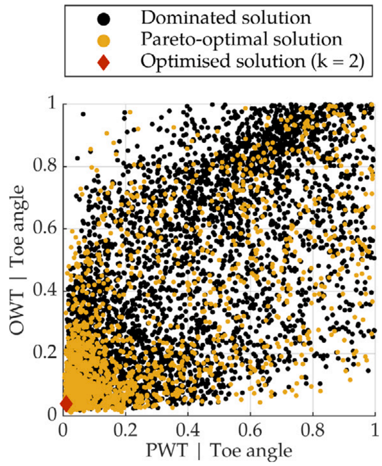

Fig. 14. Projection onto the plane of the objective functions f_1–f_4 (see Table 7) of the nine-dimensional CD+ set (see Section 5.2). Pareto-optimal solutions and the solution selected to represent MS method in Fig. 11 are highlighted.

Table 11
KEMOGA parameters.

Parameter	Value
Population size	200
Maximum generations	40 000
Number of bits	8
Mutation probability	$1 \cdot 10^{-1}$

- if the individual is not Pareto-optimal or if one of its HPs lie outside the envelope of the spheres, the fitness value is set to zero;
- otherwise, the fitness values equals $k + 1$, where k denotes the k-optimality level of the configuration (see Section 2.2).

Subsequently, individuals are paired with a probability proportional to their fitness, and single-point crossover is applied: parent chromosomes are cut at the same position and the resulting segments are swapped. The number of crossover events per generation is set to half the population size, rounded down. Following crossover, offspring are subjected to mutation, where each bit may be altered with a given probability. At the end of each generation, duplicates are removed, individuals exceeding a fitness threshold are stored, and only the fittest ones are retained for the next generation, thereby maintaining a constant population size. The algorithm terminates once a specified number of generations has been completed. Thereafter, k-optimality is assessed among all stored configurations through FNNs forecasts, and the best 2000 configurations are selected for validation to ensure comparable computational effort with MS method. For equal k-optimality levels, configurations minimising the sum of objective functions are chosen. Relevant KEMOGA parameters are summarised in Table 11.

To assess the performance of MS method against PSI method and KEMOGA, the configurations from CD (see Section 2.1) and the two comparative methods are merged into a set denoted CD+, from which a new Pareto-optimal set is derived. Fig. 10 illustrates the number of optimal solutions in CD+, grouped by k-optimality, while Table 12 provides a comparison of the methods' performance. Notably, only MS method was able to produce 2-optimal solutions.

5.3. Characteristics of the optimised suspension

Table 9 reports the hard point locations of one of the 2-optimal suspension configuration identified at the end of the six iterations of Moving Spheres method. The column “distance” shows the Euclidean norm between final and initial HP positions (see Table 3). Since the spherical neighbourhoods used during the iterations had a fixed radius of 20 mm (see step I in Section 4), it follows that hard points 1, 2, and 4 required at least four MS method iterations to be reached. Table 10 lists the scaling factor values for the optimal configuration, along the three local axes of the bushings.

Table 12
Summary of the computational effort, number of configurations examined, and optimality levels for MS method, PSI method and KEMOGA. Simulations were performed on a laptop equipped with an Intel® Core™ i7-7700HQ (2.80–3.80 GHz), 32 GB RAM, and an NVIDIA GeForce GTX 1050 GPU (2 GB).

	MS	PSI	KEMOGA
Computational time			
Training set generation	110 h	–	110 h
FNN training	24 h	–	65 h
FNN querying	47 h	–	10 h
Data validation	28 h	216 h	16 h
Total	**209 h**	**216 h**	**201 h**
Configurations examined			
By FNN forecasts	$1.33 \cdot 10^9$	–	$8 \cdot 10^6$
By simulations – training set	9000	–	9000
By simulations – validation set	3000	25 000	2000
Optimality of the solutions			
$k = 2$	43	0	0
$k = 1$	288	7	1747
$k = 0$	2344	4.444	246

Fig. 11 illustrates the geometric modifications to the suspension. In particular, the lower control arm is stretched longitudinally, with the front point moved forward by more than 50 mm, and lowered closer to the ground. The height difference between front and rear points is reduced – from 15 mm in the original configuration to about 3 mm – while the vertical offset between the outer point and the other two exceeds 30 mm. The tie rod, instead, is significantly extended laterally, with its length increased by approximately 30%, and a reduced longitudinal distance between endpoints, resulting in a smaller angle relative to the lateral axis.

The elasto-kinematic behaviour of the optimised suspension is shown in Fig. 12. Acceptance bounds are shown in red, the dashed line represents the original suspension, whereas one optimised configuration per method is displayed, each in a different colour. Toe angle in PWT varies linearly under positive wheel travel (bump), as its curvature was included as an objective function. All other metrics are improved with respect to the reference design, with most closely aligned to the midline of the bounds. The optimal solution from MS method shows significant improvement in toe angle across all three load cases, as well as in the camber angle response to longitudinal force at the tyre contact patch. Fig. 14 presents a projection of the nine-dimensional Pareto-optimal set onto the objective function plane f_1–f_4 (see Table 7).

6. Conclusions

Moving Spheres is a method to solve the multi-objective design optimisation of mechanical systems in which the locations of the joints in three-dimensional space are of major interest. The method has proven its validity for application to multi-body systems with a large number of design variables. In this paper, Moving Spheres method was used to optimise a suspension configuration with twenty-one design variables and nine objective functions. The optimised suspension meets all of the elasto-kinematic targets. The original reference suspension did not meet the elasto-kinematic targets.

Results of the proposed method are compared to those obtained through Parameter Space Investigation and KEMOGA, considering a similar computational effort. The comparative analysis shows that the optimal solutions found by Moving Spheres method dominate those found by PSI and KEMOGA. The optimal solutions determined in the last iteration step of Moving Spheres method are characterised by the highest optimality level, demonstrating that MS method is able to converge towards the regions of the design space where optimal solutions are located.

CRediT authorship contribution statement

Lorenzo De Santanna: Writing – review & editing, Writing – original draft, Software, Investigation, Data curation. **Massimiliano Gobbi:** Writing – review & editing, Supervision, Software, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Riccardo Malacrida:** Writing – original draft, Visualization, Software, Investigation, Data curation. **Gianpiero Mastinu:** Writing – review & editing, Supervision, Funding acquisition, Formal analysis.

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.

Acknowledgements

The authors wish to express gratitude to Daniele Catelani from Hexagon and Andrea Bianchetti from VI-Grade for their support during the simulation activities. We confirm that the manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed.

Data availability

The data required to reproduce the results are directly provided in the document.

References

[1] Statnikov Roman, Statnikov Alexander. The parameter space investigation method toolkit. Boston/London: Artech House; 2011.
[2] Steuer Ralph E, Sun Minghe. The parameter space investigation method of multiple objective nonlinear programming: A computational investigation. Oper Res 1995;43(4):641–8, URL https://www.jstor.org/stable/171690.
[3] Gobbi Massimiliano. A k, k-ε optimality selection based multi objective genetic algorithm with applications to vehicle engineering. Optim Eng 2013;14(2):345–60. http://dx.doi.org/10.1007/s11081-011-9185-8.
[4] Edgeworth Francis Ysidro. Mathematical psychics: An essay on the application of mathematics to the moral sciences. London: C. Kegan Paul & Co; 1881.
[5] Pareto Vilfredo. Manuale di economia politica. Milano: Società Editrice Libraria; 1906.
[6] Deb Kalyanmoy. Multi-objective optimization using evolutionary algorithms. Hoboken, NJ, USA: John Wiley & Sons, Inc.; 2001.
[7] Srinivas N, Deb Kalyanmoy. Multi-objective optimization using nondominated sorting in genetic algorithms. Evol Comput 1994;2(3):221–48. http://dx.doi.org/10.1162/evco.1994.2.3.221.
[8] Zitzler Eckart, Deb Kalyanmoy, Thiele Lothar. Comparison of multiobjective evolutionary algorithms: Empirical results. Evol Comput 2000;8(2):173–95. http://dx.doi.org/10.1162/106365600568202.
[9] Deb Kalyanmoy, Pratap Anand, Agarwal Sameer, Meyarivan T. A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 2002;6(2):182–97. http://dx.doi.org/10.1109/4235.996017.
[10] Coello Carlos Artemio Coello. Evolutionary algorithms for solving multi-objective problems. New York: Springer; 2007.
[11] Levi Francesco, Gobbi Massimiliano, Farina Marco, Mastinu Giampiero. Multi-objective design and selection of one single optimal solution. 2004, http://dx.doi.org/10.1115/IMECE2004-60902.
[12] Papaioannou Georgios, Koulocheris Dimitrios. Multi-objective optimization of semi-active suspensions using KEMOGA algorithm. Eng Sci Technol an Int J 2019;22(4):1035–46. http://dx.doi.org/10.1016/j.jestch.2019.02.013.
[13] Russo Matteo, Zhang Dan, Liu Xin-Jun, Xie Zenghui. A review of parallel kinematic machine tools: Design, modeling, and applications. Int J Mach Tools Manuf 2024;196:104118. http://dx.doi.org/10.1016/j.ijmachtools.2024.104118.
[14] Gao Zhen, Zhang Dan, Ge Yunjian. Design optimization of a spatial six degree-of-freedom parallel manipulator based on artificial intelligence approaches. Robot Comput-Integr Manuf 2010;26:180–9. http://dx.doi.org/10.1016/j.rcim.2009.07.002.
[15] Zou Qi, Zhang Dan, Luo Xueling, Huang Guanyu, Li Lijian, Zhang Haiqiang. Enumeration and optimum design of a class of translational parallel mechanisms with prismatic and parallelogram joints. Mech Mach Theory 2020;150:103846. http://dx.doi.org/10.1016/j.mechmachtheory.2020.103846.
[16] Huang Guanyu, Zhang Dan, Zou Qi, Ye Wei, Kong Lingyu. Analysis and design method of a class of reconfigurable parallel mechanisms by using reconfigurable platform. Mech Mach Theory 2023;181:105215. http://dx.doi.org/10.1016/j.mechmachtheory.2022.105215.
[17] Zhang Dan, Gao Zhen, Fassi Irene. Design optimization of a spatial hybrid mechanism for micromanipulation. Int J Mech Mater Des 2011;7(1):55–70. http://dx.doi.org/10.1007/s10999-011-9149-3.
[18] Zhang Dan, Wei Bin. Optimization and stiffness performance analysis for 3-DOF spatial and spherical parallel mechanisms. In: ASME international mechanical engineering congress and exposition, Volume 4A: Dynamics, Vibration, and Control, 2016, V04AT05A051. http://dx.doi.org/10.1115/IMECE2016-66682.
[19] Ling Mingxiang, Song Dezhi, Zhang Xianmin, He Xin, Li Hai, Wu Mengxiang, Cao Lei, Lu Shenglin. Analysis and design of spatial compliant mechanisms using a 3-D dynamic stiffness model. Mech Mach Theory 2022;168:104581. http://dx.doi.org/10.1016/j.mechmachtheory.2021.104581.
[20] Guacheta-Alba Juan Camilo, Nunez Diego Alejandro, Dutra Max Suell, Mauledoux Mauricio, Avilés Oscar Fernando. Multi-objective optimization of 6-DOF deposition trajectories using NSGA-II. J Braz Soc Mech Sci Eng 2023;45(11):610. http://dx.doi.org/10.1007/s40430-023-04495-1.
[21] Gu Yuping, Feng Shihao, Guo Yuqin, Wan Fang, Dai Jian S, Pan Jia, Song Chaoyang. Overconstrained coaxial design of robotic legs with omni-directional locomotion. Mech Mach Theory 2022;176:105018. http://dx.doi.org/10.1016/j.mechmachtheory.2022.105018.
[22] Shi Chuang, Guo Hongwei, Cheng Yadi, Liu Rongqiang, Deng Zongquan. Design and multi-objective comprehensive optimization of cable-strut tensioned antenna mechanism. Acta Astronaut 2021;178:406–22. http://dx.doi.org/10.1016/j.actaastro.2020.09.031.
[23] Pan Yongjun, Hou Liang. Lifting and parallel lifting optimization by using sensitivity and fuzzy set for an earthmoving mechanism. Proc Inst Mech Eng Part D: J Automob Eng 2017;231(2):192–203. http://dx.doi.org/10.1177/0954407016660454.
[24] Zhang Zi-Hui, Zhong Chong-Quan, Xu Zhi-Zheng, Teng Hong-Fei. A non-dominated sorting cooperative co-evolutionary differential evolution algorithm for multi-objective layout optimization. IEEE Access 2017;5:14468–77. http://dx.doi.org/10.1109/ACCESS.2017.2716111.
[25] Vo Nam, Tang Huy, Lee Jaehong. A multi-objective Grey Wolf–Cuckoo search algorithm applied to spatial truss design optimization. Appl Soft Comput 2024;155:111435. http://dx.doi.org/10.1016/j.asoc.2024.111435.
[26] Xue Haoxiang, Previati Giorgio, Gobbi Massimiliano, Mastinu Gianpiero. Research and development on noise, vibration, and harshness of road vehicles using driving simulators—A review. SAE Int J Veh Dyn Stab NVH 2023;7:10–07–04–0035. http://dx.doi.org/10.4271/10-07-04-0035.
[27] Mastinu Giampiero, Previati Giorgio, Rossa Fabio Della, Gobbi Massimiliano, Fainello Marco. How drivers lose control of the car. SAE Int J Veh Dyn Stab NVH 2024;8:10–08–01–0007. http://dx.doi.org/10.4271/10-08-01-0007.
[28] Mastinu Giampiero, Ploechl Manfred. Road and off-road vehicle system dynamics handbook. Boca Raton, FL, USA: CRC Press; 2014.
[29] Du Haiping, Liu Pengfei, Ning Donghong, Zhang Nong. Electrically interconnected suspension and related technologies: A comprehensive review. SAE Int J Veh Dyn Stab NVH 2023;7:10–07–03–0024. http://dx.doi.org/10.4271/10-07-03-0024.
[30] Gagliano Charles J, Gobbi Massimiliano, Mastinu Giampiero, Pennati Mario. Indoor/outdoor testing of a passenger car suspension for vibration and harshness analysis. SAE Int J Passeng Cars - Mech Syst 2012;5:2012–01–0765. http://dx.doi.org/10.4271/2012-01-0765.
[31] Gobbi Massimiliano, Guarneri Paolo, Mastinu Giampiero, Rocca Gianpiero, Castignani Luca. A method for vibration and harshness analysis based on indoor testing of automotive suspension systems. SAE Int J Mater Manuf 2010;3:2010–01–0639. http://dx.doi.org/10.4271/2010-01-0639.
[32] Derrix Daniel, Deubel Clemens, Kubenz Jan, Prokop Günther. Experimental analysis of the influence of body stiffness on dynamic suspension kinematics and compliance characteristics and dynamic body behavior. SAE Int J Veh Dyn Stab NVH 2021;5:10–05–04–0032. http://dx.doi.org/10.4271/10-05-04-0032.
[33] Yang Liunan, Ramakrishnan Kesavan, Mastinu Giampiero, Previati Giorgio, Gobbi Massimiliano. Automotive suspensions with additional spring in series with damper: Optimal design by analytical formulae. SAE Int J Veh Dyn Stab NVH 2020;4:10–04–03–0018. http://dx.doi.org/10.4271/10-04-03-0018.
[34] Yang Liunan, Gobbi Massimiliano, Mastinu Giampiero, Previati Giorgio, Ballo Federico. Multi-disciplinary optimisation of road vehicle chassis subsystems. Energies 2022;15:2172. http://dx.doi.org/10.3390/en15062172.
[35] Datoussaid Selim, Verlinden Olivier, Conti Calogero. Application of evolutionary strategies to optimal design of multibody systems. Multibody Syst Dyn 2002;8:393–408. http://dx.doi.org/10.1023/A:1021101912826.
[36] Simionescu Petru Aurelian, Beale David. Synthesis and analysis of the five-link rear suspension system used in automobiles. Mech Mach Theory 2002;37:815–32. http://dx.doi.org/10.1016/S0094-114X(02)00037-X.
[37] Choi Byung-Lyul, Yook Sunmin, Choi Dong-Hoon, Choi Jin-Ho, Kim In-Dong, Baek Hong-Jeon. The optimization of automotive suspension system considering multidisciplinary design requirements. 2009, http://dx.doi.org/10.4271/2009-01-1239.

[38] Feng Xingxing, Wu Jinglai, Zhang Yunqing, Jiang Ming. Suspension kinematic/compliance uncertain optimization using a Chebyshev polynomial approach. SAE Int J Mater Manuf 2015;8:2015–01–0432. http://dx.doi.org/10.4271/2015-01-0432.

[39] Šagi Goran. Multi-objective optimization model in the vehicle suspension system development process. Teh Vjesnik- Tech Gaz 2015;22:1021–8. http://dx.doi.org/10.17559/TV-20150220151816.

[40] Wang ChunYan, Zhang YuQi, Zhao WanZhong. Multi-objective optimization of a steering system considering steering modality. Adv Eng Softw 2018;126:61–74. http://dx.doi.org/10.1016/j.advengsoft.2018.09.012.

[41] Shi Qin, Peng Chengwang, Chen Yikai, He Jie, Li Peiqing, Chen Jiajia. Robust kinematics design of MacPherson suspension based on a double-loop multi-objective particle swarm optimization algorithm. Proc Inst Mech Eng Part D: J Automob Eng 2019;233:3263–78. http://dx.doi.org/10.1177/0954407018821556.

[42] Joo Behzad Bahrami, Jamali Ali, Nariman-zadeh Nader. Multi-objective robust design approach usage in integration of bond graph and genetic programming. Int J Modelling Simul 2022;42:743–59. http://dx.doi.org/10.1080/02286203.2021.1972788.

[43] Llopis-Albert Carlos, Rubio Francisco, Zeng Shouzhen. Multiobjective optimization framework for designing a vehicle suspension system. a comparison of optimization algorithms. Adv Eng Softw 2023;176:103375. http://dx.doi.org/10.1016/j.advengsoft.2022.103375.

[44] Hegazy Ahmed Abuelfadl, Kaldas Mina M, Soliman Aref Mohamed Ahmed, Huzayyin Ahmed Soliman. Modeling and optimization of regenerative MacPherson strut. SAE Int J Veh Dyn Stab NVH 2024;9:10–09–01–0005. http://dx.doi.org/10.4271/10-09-01-0005.

[45] Lu Ao, Li Runfeng, Yu Yunchang, Ji Wenfei, Hou Yufeng, Tian Guangyu. Impacts of dynamic toe angle variations on four-wheel independent steering control and their optimization strategies. SAE Technical Paper 2024;2024-01-2321. http://dx.doi.org/10.4271/2024-01-2321.

[46] Knapczyk Józef, Maniowski Michał. Optimization of 5-rod car suspension for elastokinematic and dynamic characteristics. Arch Mech Eng 2010;57:133–47. http://dx.doi.org/10.2478/v10180-010-0007-x.

[47] Nasiri Sayyad, Sina Naser, Eslami Abolqasem. Multi-objective optimisation of McPherson strut suspension mechanism kinematics using random search method. Indian J Sci Technol 2015;8. http://dx.doi.org/10.17485/ijst/2015/v8i16/62548.

[48] Arshad Muhammad Waqas, Lodi Stefano. Optimization of double wishbone suspension: Evaluating the performance of classes of algorithms. In: 2024 international conference on applied mathematics & computer science. ICAMCS, IEEE; 2024, p. 156–61. http://dx.doi.org/10.1109/ICAMCS62774.2024.00025.

[49] Mastinu Giampiero, Gobbi Massimiliano, Miano Carlo. Optimal design of complex mechanical systems: with applications to vehicle engineering. Berlin/Heidelberg/New York: Springer Science & Business Media; 2007.

[50] Gobbi Massimiliano, Mastinu Giampiero, Doniselli Carlo, Guglielmetto Luca, Pisino Enrico. Optimal & robust design of a road vehicle suspension system. Veh Syst Dyn Suppl 1999;33:3–22. http://dx.doi.org/10.1080/00423114.1999.12063066.

[51] Cheng Xianfu, Lin Yuqun. Multiobjective robust design of the double wishbone suspension system based on particle swarm optimization. Sci World J 2014;2014:1–7. http://dx.doi.org/10.1155/2014/354857.

[52] Miettinen Kaisa. Nonlinear multiobjective optimization. Dordrecht: Kluwer Academic Publishers; 1999.

[53] Gobbi Massimiliano, Previati Giorgio, Mastinu Giampiero. A method for the optimal design of automotive rubber components. 6, 2012, http://dx.doi.org/10.1115/DETC2012-70913,

[54] Dixon John Charles. Tires, suspension, and handling. Troy: Society of Automotive Engineers; 1996.

[55] Foresee F Dan, Hagan Martin T. Gauss-Newton approximation to Bayesian learning. Proc Int Conf Neural Netw (ICNN' 97) 1997;3:1930–5.

[56] Matsumoto Makoto, Nishimura Takuji. Mersenne twister. ACM Trans Model Comput Simul 1998;8:3–30. http://dx.doi.org/10.1145/272991.272995.