A methodology for simulation-based, multiobjective gear design optimization

Alessio Artoni[a,*]

[a] *Dipartimento di Ingegneria Civile e Industriale, University of Pisa, Largo Lucio Lazzarino 2, 56122 Pisa, Italy*

Abstract

Design optimization of geared transmissions has become more of a necessity than ever before. Typically, conflicting design goals must be concurrently achieved. The difficulty of such a multiobjective design optimization problem is exacerbated by the fact that modern design practices rely on increasingly sophisticated, computationally-expensive simulation tools for tooth contact analysis. Their intrinsic nonlinearities add complexity to the problem, hampering gear designers' efforts to obtain globally optimal solutions. Practical optimization problems of this class have often been solved by evolutionary algorithms, but their computational burden may well be inappropriate for CPU-intensive simulation models. The present work details an algorithmic framework inspired by deterministic multiobjective optimization methods, specially combined with a direct-search global optimization algorithm to obtain globally Pareto-optimal solutions. Nonlinear constraints are enforced through an exact penalty formulation. A comprehensive description of all theoretical and algorithmic details is provided, with the intention of enabling gear designers to implement or adapt the proposed methodology to their design optimization purposes. Two tests on a challenging gear design problem, namely ease-off topography optimization of a hypoid gear set for maximum efficiency and minimum contact stress, demonstrate that the proposed method can efficiently obtain solutions belonging to the global Pareto front.

Declarations of interest: none.

Keywords: gear design, gear optimization, multi-objective optimization, global optimization, simulation-based, hypoid gears

1. Introduction

A number of challenging factors, such as global competition and increasingly strict regulations, are having a significant impact on modern engineering practices. Gear design optimization has become more of a necessity than ever before. In particular,

*Corresponding author. Tel.: +39 050 2218009. Fax: +39 050 2218065
Email address: alessio.artoni@unipi.it (Alessio Artoni)

a strong emphasis is being placed on *multiobjective* gear design optimization (e.g., [1, 2, 3, 4]), as typical design problems have a number of *conflicting* objectives (cost or objective functions to be concurrently minimized or maximized). In complex system design, multiobjective optimization approaches have become ubiquitous and a sine qua non for effective design solutions.

Two major requirements have to be satisfied for successful multiobjective gear design optimization: modeling of the system at hand needs to be accurate (and validated), and applied optimization methodologies have to be practical, effective, and robust. Meeting the first requirement typically involves computational physics in the form of simulation-based approaches, such as those based on finite/boundary element methods. As a consequence, the second requirement can only be fulfilled by optimization algorithms designed to cope with two relevant issues of simulation-based design, namely *computational burden* and *numerical noise*.

In terms of CPU time, the computational burden of computer-simulated tooth contact analysis (under load) can range from fractions of a second to hours. As a result, practicable optimization strategies would benefit from computationally efficient algorithms that keep the number of function evaluations to a minimum.

Numerical noise typically appears in the form of *discretization error*. Numerical methods implemented in most simulation tools are based on different levels of discretization (e.g., the finite element mesh) that result in the output functions being discontinuous and/or having discontinuous derivatives. With nonsmooth functions, derivative-based optimization methods, i.e. all methods requiring accurate evaluation of the gradient and Hessian of the (underlying possibly smooth) objective functions, may easily fail.

The vast majority of gear design optimization problems also include *constraints* that determine a feasible region in the design variable space. In their most general form, they appear as nonlinear constraints (equalities and inequalities), involving the design variables in a nonlinear and often complicated fashion. They also appear in the form of bound constraints, which define an admissible (hyper)box containing all practical values the design variables may take on. Clearly, optimization algorithms at issue need to be able to handle such constraints.

Multiobjective optimization (*MOO*) methods have received a great deal of attention over the last few years. A distinction can be made between *classical methods*, mostly deterministic, and *evolutionary methods*, metaheuristic algorithms based on stochastic operators. Textbooks [5] and [6] are great references for the two classes. Reference [7] provides a detailed review of the state of the art in MOO methods. Evolutionary methods based upon genetic algorithms, such as NSGA-II [8] and SPEA2 [9], have proved to be effective in tackling various MOO problems. Since they do not need derivatives, they can easily deal with nonsmooth objective functions/constraints. An initial population of individuals (future solutions) is defined that progress generation after generation (increase their fitness) as a result of stochastic operators, and they can effectively obtain Pareto-optimal solutions with a good spread over the Pareto front. Despite their often excellent performance, their computational cost may be prohibitive in terms of function evaluations if one seeks *accurate optima*, particularly when nonlinear constraints are present: as an example, the number of objective function evaluations in [4] ranges from a minimum of 50000 up to 30 million (the objective functions and the constraints

therein are simple, closed-form analytical expressions, and not the result of computer simulations). Interesting, further confirmations in this sense are given by the recent works [10, 11, 12] on MOO of geared transmissions, where up to tens of thousands of function (fitness) evaluations are required for convergence. In addition to the fact that good performance often requires a proper tuning of the algorithm, which is usually problem-dependent, how to select principled termination conditions for MOO evolutionary methods is not straightforward: they are often stopped once a certain number of generations (or CPU time) have been reached. As a consequence, the search process may be stopped prematurely, and the final solutions, albeit optimal with respect to the final population, do not necessarily belong to the global, or even local, Pareto-optimal set. Or, on the other hand, wasteful extra-iterations are run after convergence has actually been achieved.

For the above reasons, the framework described in this paper draws upon classical deterministic methods. For the sake of clarity, it is worth specifying that the method proposed here is not appropriate for problems where a function evaluation requires several hours or days of CPU time: in such cases, viable approaches may be offered by *metamodeling techniques* (response surfaces, kriging models, artificial neural networks, etc.), which aim to approximate the computationally expensive simulation models through data-driven surrogate models that are cheaper to evaluate. As can be understood from [13] and [14] (to name but a few), the usage of surrogate models appears to be currently hampered by a number of factors, such as difficulties in selecting the appropriate form of the model, in estimating its parameters, and in assessing its resulting accuracy (model validation). However, recent research on metamodeling has identified some promising approaches, as discussed in [15]. A recent example of surrogate-assisted evolutionary algorithms for gear MOO is offered by Korta and Mundo [16].

This paper proposes a deterministic approach to solve simulation-based, multiobjective gear design optimization problems, in the presence of general nonlinear constraints on the design variables. Besides gear design, engineering problems such as size and shape optimization fit naturally into this framework. The proposed formulation, originally conceived to solve the gear optimization problems described in [17, 18], is geared toward computational efficiency, accuracy, and global optimality. The present contribution expands on such works by providing the gear community with all theoretical aspects, relevant references, and improved algorithmic details, with the intention of enabling interested readers, particularly gear designers, to implement or adapt the proposed method to their design optimization purposes. In addition, constraint handling has been also improved and generalized. This method is particularly tailored for nonlinear, nonsmooth problems with a low to medium number of design variables, where engineering expertise/know-how is available to specify reasonable design goals, practical resolution of the design variables as well as operational upper and lower bounds on them. The design variables are assumed to be continuous. The performance of the proposed method on a challenging gear design problem, namely MOO of the microgeometry of a hypoid gear drive, will also be shown.

2. Problem Description and Formulation

2.1. The MOO problem

The general constrained MOO problem considered in the present work can be expressed as

$$\begin{aligned} &\underset{\mathbf{x}}{\text{minimize}} \;\; \mathbf{f}(\mathbf{x}) \\ &\text{subject to} \;\; \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}, \quad \mathbf{g}(\mathbf{x}) \geq 0, \quad \mathbf{h}(\mathbf{x}) = 0 \end{aligned} \tag{1}$$

where the symbols denote the following vectors

$$\begin{aligned} &\mathbf{f} = (f_1, \ldots, f_m) \in \mathbb{R}^m, && \text{(conflicting) objective functions} \\ &\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n, && \text{design variables} \\ &\mathbf{l} = (l_1, \ldots, l_n) \in \mathbb{R}^n, && \text{lower bounds for } \mathbf{x} \\ &\mathbf{u} = (u_1, \ldots, u_n) \in \mathbb{R}^n, && \text{upper bounds for } \mathbf{x} \\ &\mathbf{g} = (g_1, \ldots, g_p) \in \mathbb{R}^p, && \text{inequality constraints} \\ &\mathbf{h} = (h_1, \ldots, h_q) \in \mathbb{R}^q, && \text{equality constraints} \end{aligned}$$

The set of constraints $(\mathbf{l}, \mathbf{u}, \mathbf{g}, \mathbf{h})$ defines the *feasible region* F. Problem (1) can be rewritten in a more compact form as

$$\min_{\mathbf{x} \in F} \mathbf{f}(\mathbf{x}) \tag{2}$$

(If some objectives $f_i(\mathbf{x})$ must be maximized one simply needs to replace f_i with $-f_i$.)

We always assume here that bound constraints $\mathbf{l}$ and $\mathbf{u}$ are present, since in real-world applications:

- The design variables typically vary within limited ranges. As an example, hardware constraints in machine tools restrict machine setting variations.
- For most global optimization algorithms, computational efficiency benefits from bound constraints as they limit the solution space to be searched.
- From a more mathematical point of view, bound constraints can act as a safeguard for optimization algorithms by preventing them from executing indefinitely when an objective function is unbounded below.

Let us now direct our attention to the objectives $\mathbf{f}$. As they generally conflict with each other, MOO problems typically have infinitely many solutions, called *Pareto-optimal* solutions (also named trade-off, compromise, non-dominated, efficient, and non-inferior solutions). Ultimately, which solution to select depends on some sort of *preference* expressed by the designer/decision maker. An MOO problem with $n = 3$ and $m = 2$ is shown in Fig. 1 (it can be extended to any dimension). The *feasible region* F is a subset of the *decision variable space* $\mathbb{R}^3$, and it is defined by the problem constraints. The objectives $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}))$, with $\mathbf{x} = (x_1, x_2, x_3)$, map F into the so-called *feasible objective region* Z ($\subset \mathbb{R}^2$). The *Pareto front*, the solid black curve

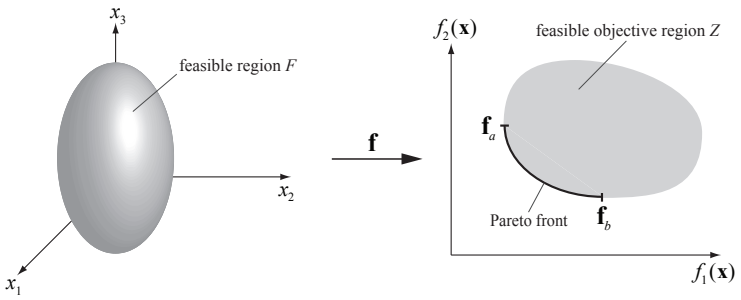

Figure 1: Example of an MOO problem (n = 3, m = 2).

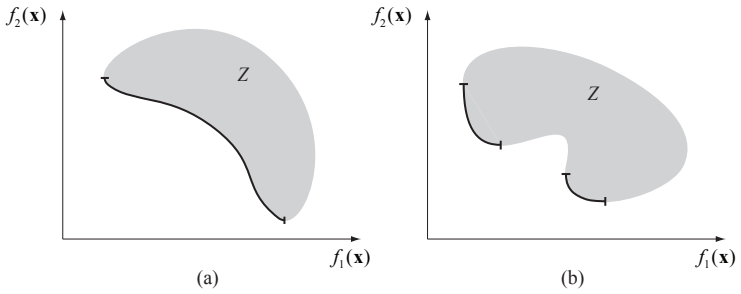

Figure 2: Examples of (a) a nonconvex, connected Pareto front and (b) a disconnected Pareto front.

between $\mathbf{f}_a$ and $\mathbf{f}_b$ in Fig. 1, is the locus of all the compromise (Pareto-optimal) solutions for which one cannot further decrease one objective function without inevitably increasing the other one.

The general nonlinear MOO problem can have convex or nonconvex objective functions and constraints, with nonconvexity being typically the rule in complex simulation-based applications. Nonconvex problems have local minima, which are reflected as *locally* (as opposed to *globally*) Pareto-optimal sets in the objective space. Globally and locally Pareto-optimal sets coincide only if the problem is convex. Obviously, one seeks globally Pareto-optimal solutions, but points of locally Pareto-optimal sets are often the computationally available solutions, since local minima "trap" most classical optimization algorithms. An exception is offered by *global* optimization algorithms, and one of them will be described and adopted in the present work.

Nonconvexity has another important meaning in MOO: it refers to the shape of the Pareto-optimal set, as illustrated in Fig. 2(a). Nonconvexity can also result in a *disconnected* Pareto front, as shown in Fig. 2(b). Nonconvex Pareto fronts can cause certain MOO algorithms to obtain solutions only in restricted regions of such fronts, missing other interesting or important solutions, and thus an algorithm that can handle nonconvexity should always be adopted.

2.2. Solution of MOO problems by classical methods

As stated in the Introduction, the framework of this work is based on deterministic, classical MOO methods, in an attempt to keep the number of function evaluations

to a minimum. Theoretical foundations of these methods, optimality definitions and conditions, duality results, as well as relevant theorems can be found in [5]. An in-depth theoretical account of MOO is given in [19].

As mentioned in Section 2.1, the solutions of an MOO problem are theoretically infinitely many: which solution to prefer over another depends on some measure of preference. *Preference is expressed by the gear engineer*, or more generally by a so-called decision maker (DM), who is ultimately responsible for the decision on the final solution.

Most classical methods are based on the concept of *scalarization*, which is the process of transforming an MOO problem into an equivalent single-objective optimization problem with a scalar-valued objective function, termed *scalarizing function*. A single Pareto-optimal solution can be obtained by minimizing the selected scalarizing function. Basically, the choice of the scalarizing function is what characterizes the different methods. While the interested reader is referred to [5, 7, 19] for taxonomies and comprehensive descriptions of classical methods, just two of them will be discussed here: the all too popular *weighting method* and the *achievement function approach*, which will be our final choice.

2.2.1. Weighting method.

The simplicity of this method makes it a very popular one. People often use it without necessarily recognizing it as an MOO method. It is based on a linear combination (weighting) of the m objective functions. The problem to be solved is

$$\min_{\mathbf{x}\in F} \left(W(\mathbf{w};\mathbf{x}) = \sum_{i=1}^{m} w_i f_i(\mathbf{x}) \right) \tag{3}$$

where the weights $\mathbf{w} = (w_1, \ldots, w_m)$ are selected such that $w_i \geq 0$ (for $i = 1, \ldots, m$) and $\sum_{i=1}^{m} w_i = 1$. Multiple solutions are generated by altering the weights. Such solutions are guaranteed to be Pareto-optimal if the weights are strictly positive.

In order to avoid that different magnitudes and/or ranges of the objectives misdirect the solution process, it is always advisable to *normalize* the objective functions according to

$$f_i^{(\mathrm{n})} = \frac{f_i - f_i^{(\mathrm{id})}}{f_i^{(\mathrm{nad})} - f_i^{(\mathrm{id})}} \tag{4}$$

where $f_i^{(\mathrm{id})}$ and $f_i^{(\mathrm{nad})}$ are, respectively, (estimates of) the components of the *ideal* and *nadir* objective vectors[1]. This reformulation is called *normalization* and provides important computational (and theoretical) advantages.

Although the weighting method is widely used to solve MOO problems, not all users are aware that this method has a serious shortcoming: a subset of Pareto-optimal solutions of *nonconvex* problems cannot be obtained, no matter how the weights are chosen. This point can be easily taken with the help of Fig. 3, where a problem with a

[1] These are the objective vectors containing, respectively, the lower and upper bounds of the Pareto front. The reader is referred to [5, pp.15–18] for a discussion on how to obtain them.

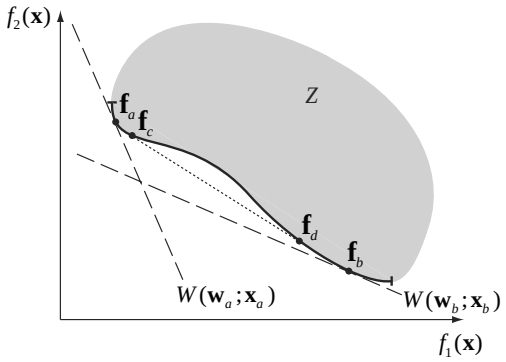

Figure 3: The weighting method on a nonconvex problem.

nonconvex Pareto front is illustrated (the two objectives are normalized). The Pareto-optimal solutions $\mathbf{f}_a$ and $\mathbf{f}_b$ can be obtained when the weights $\mathbf{w}_a = (0.7, 0.3)$ and $\mathbf{w}_b = (0.2, 0.8)$ are used, respectively, in problem (3). However, any attempt to generate any solution in the nonconvex portion of the Pareto front, i.e. between $\mathbf{f}_c$ and $\mathbf{f}_d$, would be unsuccessful. This is the major drawback of this method since it is generally difficult to check real problems for convexity, especially if they are based, as in the present scenario, on some black-box simulation model.

DMs often use the weighting method specifying a weighting vector $\mathbf{w}$ to weigh the objectives differently. However, it can be proved that specifying sensible weights is not an easy task, especially if some correlation between objectives is present.

2.2.2. Achievement function approach.

Achievement functions are scalarizing functions based on a discretionary *reference point* $\tilde{\mathbf{f}}$ specified by the DM, i.e. a set of ideal values (called *aspiration levels*) for the objective functions at hand. As is well known in the MOO literature, the basic idea of this method is to project $\tilde{\mathbf{f}}$ onto the Pareto front (cf. Fig. 4): this can be done by minimizing the selected achievement function s: $(F \subset \mathbb{R}^n) \to \mathbb{R}$ as per

$$\min_{\mathbf{x}\in F} s(\mathbf{f}(\mathbf{x}), \tilde{\mathbf{f}}) \tag{5}$$

where $\tilde{\mathbf{f}}$ is the arbitrary reference point of aspiration levels, which can be *feasible or infeasible* (i.e., inside or outside the feasible objective region Z).

An important class of achievement functions is that of *penalty scalarizing functions*. Interesting insights into them are provided by Wierzbicki [20]. A differentiable penalty scalarizing functions is

$$s_2(\mathbf{f}(\mathbf{x}), \tilde{\mathbf{f}}; \rho) = -\left\|\mathbf{f}(\mathbf{x}) - \tilde{\mathbf{f}}\right\|^2 + \rho\left\|(\mathbf{f}(\mathbf{x}) - \tilde{\mathbf{f}})_+\right\|^2 \tag{6}$$

where $\rho > 1$ is the *penalty coefficient*, the operator $\|\cdot\|$ denotes the Euclidean norm, and the expression $(\mathbf{f}(\mathbf{x}) - \tilde{\mathbf{f}})_+$ indicates the vector whose ith element is

$$(f_i(\mathbf{x}) - \tilde{f}_i)_+ = \max\,(0, f_i(\mathbf{x}) - \tilde{f}_i(\mathbf{x})), \qquad i = 1, \ldots, m \tag{7}$$

For computational purposes, the normalized version of (6) should be used (as done in [17]), that is

$$s_2(\mathbf{f}(\mathbf{x}), \tilde{\mathbf{f}}; \rho, \boldsymbol{\omega}) = -\sum_{i=1}^{m}\Big(\omega_i(f_i(\mathbf{x}) - \tilde{f}_i)\Big)^2 + \rho\sum_{i=1}^{m}\max\Big(0, \omega_i(f_i(\mathbf{x}) - \tilde{f}_i)\Big)^2 \tag{8}$$

where the coefficients $\omega_i > 0$ are *fixed*, i. e. they are not used as weights but as normalizing coefficients: along the lines of Eq. (4), one should use

$$\omega_i = (f_i^{(\mathrm{nad})} - f_i^{(\mathrm{id})})^{-1} \tag{9}$$

Another non-differentiable but better-conditioned penalty scalarizing function is (see also [18])

$$s_\infty(\mathbf{f}(\mathbf{x}), \tilde{\mathbf{f}}; \rho, \boldsymbol{\omega}) = \max_{i=1,\ldots,m}\Big(\omega_i(f_i(\mathbf{x}) - \tilde{f}_i)\Big) + \rho_\infty\sum_{i=1}^{m}\Big(\omega_i(f_i(\mathbf{x}) - \tilde{f}_i)\Big) \tag{10}$$

where $\rho_\infty > 0$ (e.g., $\rho_\infty = 10^{-4}$) is a small augmentation coefficient required to avoid weakly Pareto-optimal solutions. Minimization of (8) or (10) yields a Pareto-optimal solution. The obtained solution technically depends on the selected metric, on the coefficients ω_i, and on the chosen value of the penalty coefficient ρ, but the important theoretical dependence is on the reference point used. The fact remains, however, that such technical dependencies can make a difference in terms of computational performance. The achievement function s_∞ will be preferred to s_2 hereafter.

It is important to highlight that, unlike other MOO methods, the achievement function approach projects both feasible and infeasible reference points onto the Pareto front, as depicted in Fig. 4. Nonconvex problems can be easily handled by the method. Unlike *goal programming* methods [5, 121–129], which cannot improve on feasible (i.e., pessimistic) reference points, the achievement function approach can obtain a Pareto-optimal solution even when a strictly feasible reference point is specified. In general, achievement functions are very appealing for generating Pareto-optimal solutions, since they can overcome most of the limitations associated with other methods of this category. We finally selected this approach to avoid potentially dangerous pitfalls of other classical MOO methods, which can be summarized as follows:

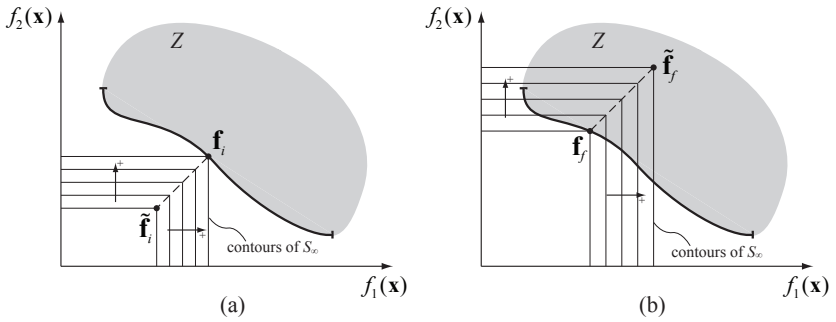

Figure 4: A nonconvex MOO problem tackled by the achievement function approach: projection of an infeasible reference point $\tilde{\mathbf{f}}_i$ (a) and of a feasible reference point $\tilde{\mathbf{f}}_f$ (b) (with $\omega_i = 1$).

- The popular weighting method and its variants are not able to obtain solutions that belong to nonconvex regions of the Pareto front.
- Other methods are not able to improve on pessimistic (i.e., feasible) reference points: this is the case, for instance, of the well-known goal programming method.
- With weighting metrics methods, specifying sensible values for the weighting coefficients is intuitive only with linear MOO problems. In general, it is likely that the intention expressed by the DM through the weighting coefficients is not met in nonlinear problems.
- Some other methods, namely the ϵ-constraint method and its variants, require a large amount of calculations to represent the Pareto-optimal set, which is in contrast with our demand for computational efficiency.

2.3. Exploration of the Pareto front: the reference point method

Once a Pareto-optimal solution is obtained, the DM could be interested in exploring the Pareto front around that solution, particularly in those cases where the DM's background knowledge is insufficient to express a sensible, informed reference point. To that end, the *reference point method* ([5, pp. 164–170], [7, pp. 38–45], and references therein) can be employed to obtain and compare a number of nearby trade-off solutions. As the name suggests, this method is based on reference points, feasible or infeasible, which are reasonable or desirable to the DM. Combination with the achievement function approach is straightforward, as can be noted by considering the following basic steps of the method:

1. Set $h = 0$ (iteration counter).
2. Ask the DM to specify a reference point $\tilde{\mathbf{f}}_h \in \mathbb{R}^m$.
3. Minimize the selected achievement function (problem (5)) to obtain its corresponding Pareto-optimal objective vector $\mathbf{f}_h = \mathbf{f}(\mathbf{x}_h)$, where $\mathbf{x}_h$ is its Pareto-optimal solution.
4. Obtain m new Pareto-optimal solutions using the following reference points in problem (5)
$$\tilde{\mathbf{f}}_{h_i} = \tilde{\mathbf{f}}_h + d_h \mathbf{e}_i, \qquad i = (1, \ldots, m) \tag{11}$$
where $d_h = \left\| \tilde{\mathbf{f}}_h - \mathbf{f}_h \right\|$ and $\mathbf{e}_i$ are the unit vectors marking the axes of the objective space.
5. If the DM is satisfied with one of the $m + 1$ solutions, stop. Otherwise, the DM selects the most interesting reference point $\tilde{\mathbf{f}}_{h_i}$, then set $h = h + 1$, $\tilde{\mathbf{f}}_{h+1} \leftarrow \tilde{\mathbf{f}}_{h_i}$, and restart from step 3.

A pictorial representation of the first iteration of the reference point method is given in Fig. 5. The ideal objective vector $\mathbf{f}^{\mathrm{id}}$ has been selected as initial reference point. The DM will then decide to proceed with $\tilde{\mathbf{f}}_1 \leftarrow \tilde{\mathbf{f}}_{0_1}$ or $\tilde{\mathbf{f}}_1 \leftarrow \tilde{\mathbf{f}}_{0_2}$, depending on which Pareto-optimal objective vector (s)he will deem more interesting between $\mathbf{f}_{0_1}$ and $\mathbf{f}_{0_2}$. Further iterations may be carried out until the DM has obtained a satisfactory solution and/or a sufficiently informative representation of the Pareto front.

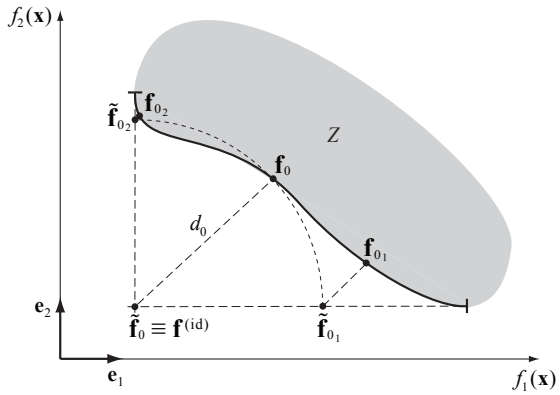

Figure 5: First iteration of the reference point method on an MOO problem with two objectives.

Figure 5 highlights the fact that the reference point method enables the DM to gain a clearer conception of the Pareto front and of the possible solutions: when the DM selects a reference point that is distant from the front, (s)he gets a wider representation of the front itself, and as the distance of the reference point decreases, a progressively finer (local) representation is obtained.

2.4. Dealing with constraints

The feasible region F is determined by the MOO problem's constraints. The solution of problem (5) requires some strategy to handle the general nonlinear constraints $\mathbf{g}(\mathbf{x})$ and $\mathbf{h}(\mathbf{x})$ (the simple bounds $\mathbf{l} \leq \mathbf{x} \leq \mathbf{u}$ will be dealt with later).

As discussed in [21, chap. 6] and [22, chap. 17], one could solve optimization problems with nonlinear constraints by formulating and solving a sequence of unconstrained problems (ideally only one) that would eventually converge to a feasible solution. (Please replace "unconstrained" with "bound-constrained" in this context.) One of the best known methods belonging to this category is the *penalty function method*, and the *quadratic penalty function* [21, sect. 6.2.1.1] is very popular and could be applied to solve problem (5) according to the following formulation

$$\begin{aligned} \min_{\mathbf{x}} \quad & s_\infty(\mathbf{f}(\mathbf{x}), \tilde{\mathbf{f}}; \rho, \boldsymbol{\omega}) + \frac{\mu}{2}\Big[\sum_{j=1}^{q} h_j^2(\mathbf{x}) + \sum_{k=1}^{p}(\max(-g_k(\mathbf{x}), 0))^2\Big] \\ \text{s. t.} \quad & \mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \end{aligned} \tag{12}$$

The non-negative *penalty parameter* μ (not to be confused with the penalty coefficient ρ in Eq. (6)) penalizes constraint violations. Formulation (12) transforms the original constrained optimization problem into a simpler bound-constrained one. However, the problem with differentiable penalty functions is that μ should be increased theoretically to infinity to insure that the minimizer of problem (12) coincides with that of the original problem. In practice, a sequence of bound-constrained problems (12) need be solved, with increasing values of the penalty parameter, until constraints are

"sufficiently" satisfied. The main drawback of such a method is that it suffers from inevitable ill-conditioning, with problem (12) becoming increasingly badly-conditioned as μ increases. The most evident consequences of ill-conditioning are deterioration in accuracy, increased computational effort, and even risk of failure of the optimization algorithm.

The problems described for differentiable penalty functions are analogous to those affecting differentiable *logarithmic barrier functions* [21, sect. 6.2.1.2]: even for smooth, well-posed problems, these methods suffer from ill-conditioning and the need to solve a sequence of subproblems, thereby placing an extra burden on computational resources. Furthermore, considering the present scenario, there is no need to insist on a differentiable penalty function if the model functions at hand are nonsmooth in the first place.

An alternative, non-differentiable, but well-conditioned approach is offered by the *absolute value penalty function* [21, sect. 6.2.2], by which problem (12) can be reformulated as

$$\begin{aligned} \min_{\mathbf{x}} \quad & s_\infty(\mathbf{f}(\mathbf{x}), \tilde{\mathbf{f}}; \rho, \omega) + \mu\left[\sum_{j=1}^{q}\left|h_j(\mathbf{x})\right| + \sum_{k=1}^{p}\max(-g_k(\mathbf{x}), 0)\right] \\ \text{s. t.} \quad & \mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \end{aligned} \tag{13}$$

The key distinction between the quadratic penalty function in Eq. (12) and the absolute value penalty function in Eq. (13) is that, under mild conditions, μ does not need to grow indefinitely, which makes the problem less ill-conditioned: for *any* $\mu > \bar{\mu}$, the solution of problem (13) coincides with the solution of the original problem. Unfortunately, $\bar{\mu}$ is not known in general, therefore it must be estimated and increased if necessary (see Section 3).

Lastly, let us point out that penalty functions may be unbounded below. However, as mentioned in section 2.1, bound constraints act as a safeguard for the optimization problems under consideration: by restricting the search region to a specific subset of $\mathbb{R}^n$, they prevent optimization algorithms from running indefinitely.

Several other methods exist to handle nonlinear constraints, and the interested reader is referred to [21] and [22] for detailed and comprehensive descriptions.

3. Solution Method and Algorithms

With section 2.1, the groundwork has been laid for the solution process of general simulation-based MOO problems. A computational framework that integrates the described techniques is sketched in Fig. 6. What has not yet been discussed is how to practically solve the central, single-objective optimization problem (13).

3.1. Main requirements

Let us recall and enumerate the main properties of the functions involved (see also [23], which presents an interesting review of some direct-search methods):

1. The objective functions $\mathbf{f}$ in the achievement function s as well as the constraints $\mathbf{g}$ are not available analytically. They are only computable, since they exist only numerically through the simulation model at hand.

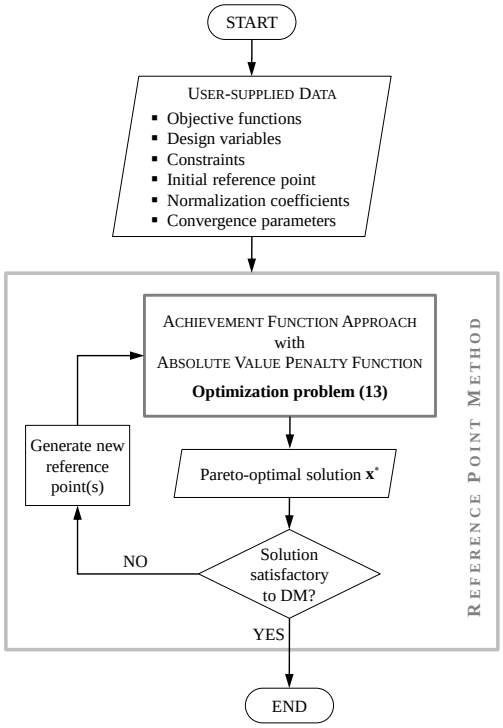

Figure 6: Flowchart of the proposed optimization framework.

2. In the simulation-based optimization setting, the functions and constraints resulting from computer simulations and their post-processing typically suffer from discretization error and numerical noise, which causes the calculation of partial derivatives for gradient-based optimization algorithm quite impractical, even for underlying smooth problems. Finite-difference approximations of the gradient and Hessian would be generally unreliable.
3. Typically, the objective function of problem (13) has a number of *local minima*, corresponding to locally Pareto-optimal solutions. Ideally, however, one seeks *global minima*, corresponding to globally Pareto-optimal solutions. This is indeed a much harder task, but classical single-objective optimization algorithms may be easily trapped into local, and potentially infeasible, solutions. Obtaining global optima is also important for a correct application of the reference point method: comparing different trade-off solutions is meaningless when some of them are just locally Pareto-optimal (hence, dominated). This fact is often not adequately stressed in classical MOO literature. A *global optimization* algorithm is needed.
4. The optimization algorithm used to solve problem (13) must be able to handle bound constraints.

Summarizing, the solution of problem (13) calls for a derivative-free, global optimization algorithm able to handle noisy, nonsmooth functions, with bound constraints on the design variables. State-of-the-art research has yielded a plethora of methods meeting the above-listed requirements [24, 25, 26]. Again, for the reasons discussed in the preceding sections, evolutionary and metamodeling techniques are excluded from consideration in the present setting.

Based on several reviews, such as those just cited, the *DIRECT* algorithm [27] and Gablonsky's implementation of it [28], called *DIRECT-1*, were deemed appropriate for the category of optimization problems at issue. However, it should be brought to the reader's attention that other algorithms (individually or possibly combined into hybrid strategies) may be better suited to specific problem types. For instance, Huyer and Neumaier's multilevel coordinate search (MCS) method [29] has proven successful and computationally efficient on a number of smooth global optimization problems, and it should be tried whenever the problem functions are basically smooth. Again, the reader should be exposed to the algorithmic performance trends highlighted in [24, 25, 26] and the references therein.

3.2. The DIRECT algorithm

The theoretical background of the DIRECT algorithm is presented in [27], and a description of an extended version of it (which can handle inequality and integer constraints) is offered by Jones in [30]. The locally-biased form of it, DIRECT-1, is described by Gablonsky in [28]. Sequential and parallel Fortran implementations of DIRECT exist, and some of them can be freely downloaded from the Web, where free implementations of the *implicit filtering* method can also be retrieved: the latter is reported to be effective when problem (13) is known to have noise-induced spurious local minima, but an underlying smooth objective function with a single global solution. Commercial versions of DIRECT, reportedly more efficient than the freely available implementations, are included in the TOMLAB optimization environment (in Matlab).

The DIRECT (DIviding RECTangles) algorithm is a derivative-free pattern search method that attempts to efficiently find a global optimizer of bound-constrained optimization problems. It is designed to completely explore the variable space, even after one or more local minima have been found. DIRECT-1 is more biased toward local search, which has proved to perform better for problems with a single global minimizer and only a few local minimizers.

As described in [17], the algorithm starts with one n-dimensional hyper-rectangle (determined by the bounds on the variables) and, at each iteration, it updates a set of hyper-rectangles by partitioning *some* of its members into smaller hyper-rectangles. Each selected hyper-rectangle is trisected along one of its long sides, after which the center points of the outer thirds are sampled. The algorithm continues until specific termination criteria are met.

DIRECT-1 attempts to reduce the number of function evaluations by grouping the hyper-rectangles according to a different metric and by trisecting just one of them at each iteration. The idea is that the overall number of divisions will be reduced, and that most of this reduction will occur in the large hyper-rectangles that are not near the global optimum. The implementation described in [31] can also handle hidden

constraints on the variables, i.e. all those cases where the objective function is not defined or cannot be computed for specific values of the design variables.

DIRECT and DIRECT-1, like other global optimization methods, suffer from a curse of dimensionality that limits them to lower-dimensional problems with up to, say, 25-30 design variables.

3.2.1. Termination criteria.

A number of termination conditions are available, with two of them being appropriate for the problems under consideration: maximum number of function evaluations (`maxf`) and minimum hyper-rectangle size (`sigmaper`, designated σ_p in the following). The former is an approximate upper bound on the maximum number of function evaluations, which is suitable for computationally demanding problems when only a limited budget of function evaluations is allowed, while the latter specifies that the optimization is terminated when the hyper-rectangle containing the minimum function value has size less than σ_p. The way the size of a hyper-rectangle $\mathcal{R}$ is measured depends on the version of DIRECT: the original DIRECT uses the distance between the center of $\mathcal{R}$ and one of its vertices, while DIRECT-1 takes the length of the longest side of $\mathcal{R}$.

For the vast majority of simulation-based problems, a *resolution* value τ_i can be determined for each design variable x_i, meaning that a variation τ_i in the i-th design variable x_i yields an unimportant (or not appreciable) variation in the objective functions and constraints. For instance, let us consider a gear micro-geometry optimization problem whose design variables have linear dimensions (like ease-off topography): it is quite unlikely that variations of the order of hundredths of a micron can affect the results. Considerable savings in computational resources can be made if one correctly accounts for the resolutions of the design variables. To this end, it is important to note that both versions of DIRECT normalize each variable $\mathbf{x}_i$ between 0 and 1. Therefore, with given resolutions $(\tau_1, \ldots, \tau_n)$, the scaled resolution

$$\hat{\tau} = \min_i \left(\frac{\tau_i}{u_i - l_i} \right) \tag{14}$$

can be used to terminate the optimization process when the current hyperrectangle size drops below $\hat{\tau}$, i.e. $\sigma_p = \hat{\tau}$. In what follows, this termination condition will be adopted.

3.3. Solution process: algorithmic framework

The proposed algorithmic framework is illustrated by the pseudocode in Fig. 7, which summarizes and gathers together the topics discussed to this point.

A value $\rho_\infty = 10^{-4}$ can be used for the penalty coefficient in Eq. (10). For the penalty parameter μ (Eq. (13)), a starting value of 10 turned out to be effective when the constraint violation terms are properly scaled with respect to the achievement function. If this is not the case, it is certainly advisable to scale the constraint violation term accordingly.

INPUT: objective functions $\mathbf{f}(\mathbf{x})$, lower and upper bounds $\mathbf{l}$ and $\mathbf{u}$ on the design variables $\mathbf{x}$, constraints $\mathbf{g}(\mathbf{x})$ and $\mathbf{h}(\mathbf{x})$, initial reference point $\tilde{\mathbf{f}}_0$, normalizing coefficients ω_i, maximum number of function evaluations allowed F, maximum number of outer iterations I_o on penalty parameter μ (before regarding the problem as infeasible), resolutions τ_i of the design variables, positive feasibility tolerances ϵ_g (for inequality constraints) and ϵ_h (for equality constraints), penalty coefficient (ρ or) ρ_∞.

---

1. Set $i = 0$, $\mu = 10$, and the logical variable `feasible` = **false**.
2. Create achievement function (Eq. (10)).
3. **do while** ($i < I_o$)
   - 3.1. Set $i = i + 1$
   - 3.2. Obtain solution $\mathbf{x}^*$ of problem (13) using DIRECT, with $\sigma_p = \tilde{\tau}$ (Eq. 14) and F as termination parameters.
   - 3.3. **if** ($\min_k(g_k(\mathbf{x}^*)) \geq -\epsilon_g$ **and** $\max_j |h_j(\mathbf{x}^*)| \leq \epsilon_h$) **then**
     `feasible` = **true**
   - 3.4. **if** (F was reached but σ_p was not reached) **then**
     stop (max number of function evaluations reached)
     else if (σ_p was reached) **then**
     if (`feasible` = **false**) **then**
     $\mu = 10\mu$
     else
     exit (leave **do** loop)
     else
     stop (some problem has occurred with DIRECT)
4. **if** ($\mathbf{f}(\mathbf{x}^*)$ is satisfactory to the DM) **then**
   terminate optimization with final solution $\mathbf{x}^*$
   else
   generate m new reference points (Eq. (11)) and restart with each of them from step 1.

Figure 7: Pseudocode of the proposed algorithmic framework.

4. Numerical Tests on a Gear Optimization Problem

Two real-world applications of the proposed method for multiobjective micro-geometry optimization of spiral bevel and hypoid gears are described in the works [17, 18]. The latter also describes how to incorporate robustness into the objective functions in order to achieve robust design optimization.

In this section, a still real but more "educational" application of the method is presented. The focus here is on providing more insight into the optimization problem formulation and its solution(s).

4.1. Multiobjective micro-geometry optimization of spiral bevel and hypoid gears

Spiral bevel and hypoid gears (Fig. 8) represent a very general (and complex) type of gears. While the former (zero shaft offset) are mostly employed in aircraft and rotorcraft drive trains, the latter are extensively used in the differentials of rear- and

four-wheel drive vehicles. Automotive and aerospace industries have been continually demanding more power density and less noise from their transmissions, as well as reduced sensitivity to misalignments. Nowadays, a strong emphasis is also being placed on maximizing their mechanical efficiency.

The tooth surfaces of typical spiral bevel and hypoid gears are mismatched, i.e. nonconjugate (hence they are theoretically in point contact, instead of line contact), in an attempt to avoid edge-contact, which could be catastrophic for the gear pair. Their deviations (*ease-off*) from their conjugate counterparts are extremely small, typically tens to few hundreds microns. Contact properties are strongly affected by the tooth surface *micro-geometry*, thus it plays an important role in their optimization.

Micro-geometry optimization of gears is a multiobjective optimization problem, since common design goals are typically conflicting. It has long been a trial-and-error (and time-consuming) process, mostly entrusted to the skills of expert gear engineers. In addition, this problem has intrinsic nonlinearities that make the process nonintuitive.

In what follows, only two objectives will be considered, so that the Pareto front, reference points, and solutions can be easily displayed in the objective space. Firstly, a case with three design variables is presented, which serves mostly to "visualize" the proposed method and to demonstrate its effectiveness. Secondly, the performance of the method is observed after increasing the number of design variables to five, which corresponds to a real design case.

4.1.1. Simulation tool.

The proposed optimization procedure relies on an LTCA (loaded tooth contact analysis) simulation tool. Such LTCA program, named *HAP*, was developed by Kolivand and Kahraman [32] to assess contact properties under load, including mechanical efficiency [33]. HAP requires significantly less computational effort than models based on finite elements: on a 2.9-GHz-processor, 8-GB-RAM computer system, one full LTCA analysis requires about 8 seconds.

4.1.2. Design variables.

As described in [17], ease-off is usually represented in the gear's axial plane (r, z), called gear projection plane, where the boundary cones of both the gear and the pinion

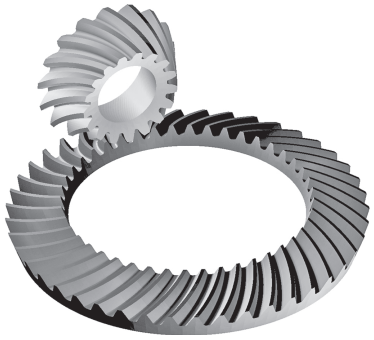

Figure 8: A hypoid gear drive.

delimit the *potential contact area* (PCA): all contact zones will lie within the PCA. The PCA can be approximated as a convex quadrilateral, and its shape and size are affected by the relative position between pinion and gear, hence by misalignments. For convenience we map the PCA to a rectangular domain, whose lengthwise and profile directions are parameterized respectively by coordinates u and v. A contour plot representation of an example ease-off surface along with a three-dimensional discrete version of it (ease-off topography) are shown in Fig. 9.

We now need to define design variables that can effectively control the shape of the ease-off surface. This can be done by modeling this surface as a suitable polynomial surface whose coefficients can be used as design variables of the optimization problem at hand. A polynomial of second degree was chosen for the present numerical tests, as it embodies the most effective tooth surface modification elements. Higher-order effects are considered in [17]. The selected polynomial ease-off p_e is represented on the PCA (u, v)-plane according to

$$p_e(\mathbf{x}; u, v) = \sum_{i=0}^{2} \sum_{j=0}^{2-i} x_{ij} u^i v^j \tag{15}$$

The five coefficients $\mathbf{x} = (x_{01}, \ldots, x_{20})$ are our design variables (x_{00} is dropped as it would imply a change in tooth thickness). Each coefficient has a direct technological implication: as an example, x_{20} and x_{02} of the quadratic terms u^2 and v^2 respectively control lengthwise and profile crowning; x_{11} quantifies flank twist. Once the optimal ease-off surface has been determined, an effective way to obtain the corresponding machine-tool settings of the hypoid generator is described in [34].

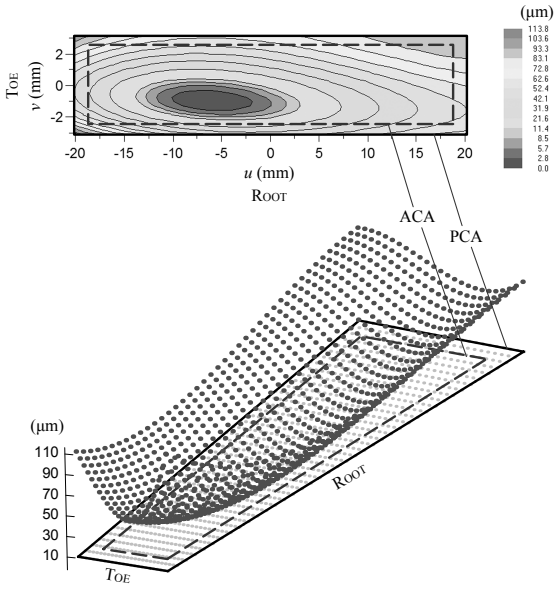

Figure 9: An example of ease-off surface on the gear-based PCA.

4.1.3. Objective functions and constraints.

Two conflicting objectives will be considered:

1. Minimization of power loss, i.e. maximization of average mechanical efficiency.
2. Minimization of maximum contact pressure.

Obviously, the second objective is pursued to prevent localized stress concentrations (and the ensuing pitting/micropitting problems). In mathematical terms, we have the two following objective functions to be concurrently *minimized*:

1. $f_1(\mathbf{x}) = W_l(\mathbf{x})$, power loss (in percentage points).
2. $f_2(\mathbf{x}) = p_{\max}(\mathbf{x})$, maximum contact pressure (in MPa).

Let us now consider constraints. To prevent edge- and corner-contact conditions, f_1 and f_2 must be minimized while avoiding that any portion of the contact pattern under load lies outside a predefined *allowable contact area* (ACA), quadrilateral (for example) in shape and, obviously, within the PCA boundaries (Fig. 9). The ACA can be specified according to the indications in [35, Annex F]. One then has to enforce that the total contact load exerted outside the ACA be zero. This can be achieved by initializing a scalar variable h to zero and augmenting its value by the (unwanted) load detected at any contact cell outside the ACA boundaries throughout one mesh cycle. This nonlinear equality constraint can be expressed by

$$h(\mathbf{x}) = 0 \tag{16}$$

where the total unwanted contact load h depends on the ease-off coefficients $\mathbf{x}$.

Finally, selecting appropriate values for $\mathbf{l}$ and $\mathbf{u}$ (bound constraints) is a simple matter. Coordinates (u, v) in Eq. (15) are rescaled so that they belong to the domain $[-1, 1] \times [-1, 1]$: the ith term $u^i v^j$ can be at most 1 (or -1), and therefore its coefficient x_{ij} coincides with the maximum (minimum) ease-off value that can be attained with that term at the PCA corners. This helps impose meaningful limits to the ease-off values. The nonlinear equality constraint (16) along with the bounds $\mathbf{l}$ and $\mathbf{u}$ determine the feasible region F.

4.1.4. Basic design data and initial reference point.

The present tests are conducted on the same automotive hypoid gear pair used in [17]. Table 1 summarizes its basic design parameters, nominal operating conditions, average mechanical efficiency η, and maximum contact pressure $p_{\max}$ calculated by HAP.

The values of η and $p_{\max}$ associated with the basic design are useful to select a valid reference point to start with. A 1% increase in η, i.e. $W_l = 4\%$, was considered ideal, as well as a value of 1100 MPa for $p_{\max}$. Therefore,

$$\tilde{\mathbf{f}}_0 = (4\%, 1100 \text{ MPa}) \tag{17}$$

was selected as initial reference point.

Basic design parameters	
Pinion tooth number	11
Gear tooth number	43
Shaft offset (mm)	30
Shaft angle (deg)	90
Transverse module (mm)	5.5
Operating conditions	
Pinion torque (Nm)	250
Pinion speed (rpm)	2000
Lubricant type	75W-90
Lubricant temperature (°C)	90
Pinion roughness R_q (μm)	1.3
Gear roughness R_q (μm)	1.7
Average mechanical efficiency η (%)	95.07
Maximum contact pressure $p_{\max}$	edge-contact

Table 1: Basic design parameters and nominal operating conditions of the hypoid gear pair [17].

4.2. Test case 1: three-variable two-objective MOO problem

This test case was created ad hoc to test the behavior of the proposed method and to facilitate its visualization. DIRECT-1 was used. The problem was set up as follows:

- Design variables $\mathbf{x} = (x_{02}, x_{11}, x_{20})$, corresponding to profile crowning, flank twist, and lengthwise crowning, respectively.
- Lower bounds $\mathbf{l} = (0.000, -0.100, 0.000)$ mm.
- Upper bounds $\mathbf{u} = (0.200, 0.100, 0.200)$ mm.
- Nonlinear equality constraint $h(\mathbf{x}) = 0$ (no contact outside the ACA, Eq. (16)).
- Design variable resolution $\tau = 0.001$ mm, corresponding to a DIRECT-1 convergence tolerance $\sigma_p = 0.005$, as per Eq. (14).

The volume (cube) defined by $\mathbf{l}$ and $\mathbf{u}$ in the design variable space was sampled at a large number of points arranged on an orthogonal grid, and an LTCA analysis was performed for each of them using HAP. For each i-th analysis, the resulting objectives $f_1(\mathbf{x}_i) = W_{l_i}$ and $f_2(\mathbf{x}_i) = p_{\max_i}$ constitute a point (objective vector) in the objective space. Also, according to Eq. (16), the value $L(\mathbf{x}_i)$ determines whether point $\mathbf{x}_i$ (decision vector) is feasible ($L(\mathbf{x}_i) = 0$) or infeasible ($L(\mathbf{x}_i) > 0$). Collecting all decision vectors $\mathbf{x}_i$ and their corresponding objective vectors $\mathbf{f}(\mathbf{x}_i)$, Figure 10 was created, where blue and red small circles denote feasible and infeasible decision vectors, respectively, as well as their corresponding feasible and infeasible objective vectors. Extracting the feasible region and the feasible objective region from it, Fig. 11 was obtained.

The application of the proposed methodology to this test case resulted in the solutions shown in Fig. 12 and Table 2. As confirmed by Fig. 12, the method was able to obtain global Pareto-optimal solutions, all of them feasible, non-dominated, and well spread over the Pareto front. Preference was intentionally given to the reference point $\tilde{\mathbf{f}}_1$ ($= \tilde{\mathbf{f}}_{0_2}$) to lead the exploration of the Pareto front toward a nonconvex region, and

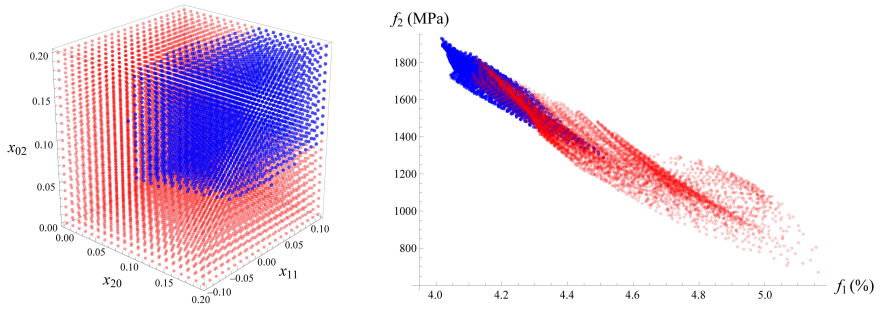

Figure 10: Test case 1: Decision variable space (left) and objective space (right). Feasible and infeasible points are denoted by blue and red small circles, respectively.

the method yielded valid solutions. It is interesting to note (Table 2) that solutions associated with notable increments in mechanical efficiency, like $\mathbf{f}_1$ and $\mathbf{f}_{1_2}$, exhibit larger contact pressures: the optimizer achieved this result by localizing the contact areas to the tooth regions where the least sliding occurred. Eventually, which solution to select depends on the DM's judgment and the specific application: just by way of example, one could choose the solution that is most robust against assembly errors or manufacturing variance.

Regarding function evaluations, each corresponding to a HAP run, their numbers are very reasonable (last column of Table 2). This attests to the computational efficiency of the proposed method.

4.3. Test case 2: five-variable two-objective MOO problem

This test case corresponds to a realistic challenging scenario where optimal first- and second-order ease-off components need to be determined. The problem was set up as follows:

- Design variables $\mathbf{x} = (x_{01}, x_{02}, x_{10}, x_{11}, x_{20})$, corresponding to pressure angle correction, profile crowning, spiral angle correction, flank twist, and lengthwise crowning, respectively.

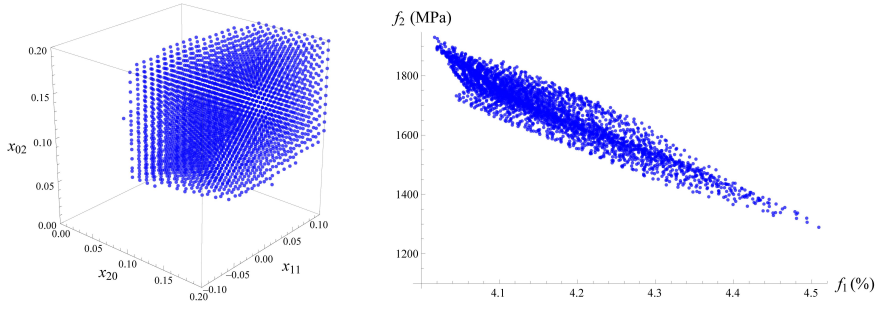

Figure 11: Test case 1: Feasible region (left) and feasible objective region (right).

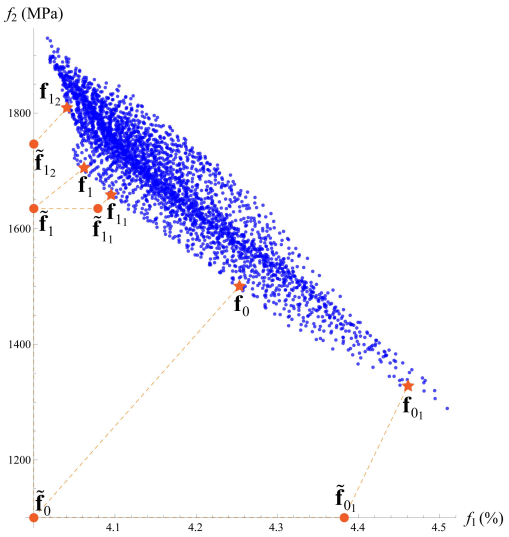

Figure 12: Test case 1: reference points and corresponding Pareto-optimal objective vectors.

- Lower bounds $\mathbf{l} = (-0.100, 0.000, -0.100, -0.100, 0.000)$ mm.
- Upper bounds $\mathbf{u} = (0.100, 0.200, 0.100, 0.100, 0.200)$ mm.
- Nonlinear equality constraint $h(\mathbf{x}) = 0$ (no contact outside the ACA, Eq. (16)).
- Design variable resolution $\tau = 0.001$ mm, corresponding to a DIRECT-1 convergence tolerance $\sigma_p = 0.005$, as per Eq. (14).

The results shown in Fig. 13 and Table 3 were obtained. Three reference points were used. Again, all solutions are non-dominated and feasible: Let us remind that a feasible solution implies that the optimal decision variables are within their bounds and that the loaded contact pattern lies inside the ACA (no edge-loading). In Fig. 13, the feasible objective region of test case 1 is also represented (grayed out) to appreciate the superior solution objective vectors obtained using five design variables (more degrees of freedom).

The three solutions obtained are very satisfactory in practical terms. The gear engineer can select the solution that is best suited to the application of the gear drive

Ref. point (%, MPa)	Sol. obj. vector (%, MPa)	Solution vector (mm)	Fun. eval.
$\tilde{\mathbf{f}}_0 = (4.00, 1100.0)$	$\mathbf{f}_0 = (4.25, 1500.6)$	$\mathbf{x}_0 = (0.1222, 0.0667, 0.1049)$	85
$\tilde{\mathbf{f}}_{0_1} = (4.38, 1100.0)$	$\mathbf{f}_{0_1} = (4.46, 1328.5)$	$\mathbf{x}_{0_1} = (0.0753, 0.0667, 0.0901)$	93
$\tilde{\mathbf{f}}_1 = (4.00, 1635.2)$	$\mathbf{f}_1 = (4.06, 1705.1)$	$\mathbf{x}_1 = (0.1938, 0.0922, 0.1551)$	121
$\tilde{\mathbf{f}}_{1_1} = (4.08, 1635.2)$	$\mathbf{f}_{1_1} = (4.09, 1658.3)$	$\mathbf{x}_{1_1} = (0.1765, 0.0971, 0.1519)$	111
$\tilde{\mathbf{f}}_{1_2} = (4.00, 1746.3)$	$\mathbf{f}_{1_2} = (4.04, 1809.2)$	$\mathbf{x}_{1_2} = (0.1996, 0.0255, 0.1560)$	117

Table 2: Test case 1: results.

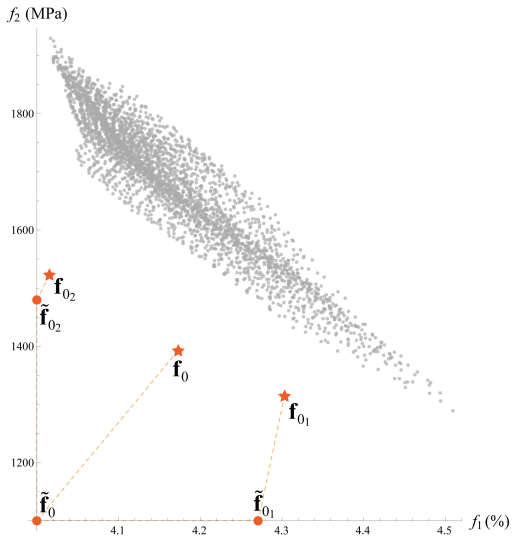

Figure 13: Test case 2: reference points and corresponding Pareto-optimal objective vectors. (Feasible objective region of test case 1 grayed out.)

at hand. Solution $\mathbf{f}_{0_1}$ offers a respectable increase in efficiency compared to the basic design conditions and a very satisfactory value of the maximum contact pressure. Solution $\mathbf{f}_{0_2}$, on the other hand, provides an excellent increase in efficiency (+1 pp) at the expense of a larger value of the maximum contact pressure. Solution $\mathbf{f}_0$ represents a nice trade-off between the other two.

In terms of function evaluations, albeit more than doubled with respect to test case 1, the results tend to attest once more that the proposed method can offer a very competitive advantage over evolutionary approaches.

5. Conclusions

With the present paper, a computational framework has been proposed to solve multiobjective gear design optimization problems, especially those whose model functions and constraints are obtained by computer-based simulation models. Special emphasis

Ref. point (%, MPa)	Sol. obj. vector (%, MPa)	Solution vector (mm)	Fun. eval.
$\tilde{\mathbf{f}}_0 = (4.00, 1100.0)$	$\mathbf{f}_0 = (4.17, 1392.8)$	$\mathbf{x}_0 = (-0.0296, 0.0926, 0.0074, 0.0700, 0.1025)$	231
$\tilde{\mathbf{f}}_{0_1} = (4.27, 1100.0)$	$\mathbf{f}_{0_1} = (4.30, 1314.2)$	$\mathbf{x}_{0_1} = (-0.0198, 0.0893, 0.0000, 0.0988, 0.1000)$	229
$\tilde{\mathbf{f}}_{0_2} = (4.00, 1479.7)$	$\mathbf{f}_{0_2} = (4.02, 1526.9)$	$\mathbf{x}_{0_2} = (-0.0749, 0.1337, -0.0181, 0.0774, 0.0918)$	311

Table 3: Test case 2: results.

has been placed on computational efficiency and on the capability of handling nonsmoothness in the problem functions.

The proposed method draws upon techniques from deterministic MOO methods, especially the achievement function approach and the reference point method, which have proved to be effective in solving multiobjective optimization problems thanks to a number of mathematical properties that other MOO methods do not possess. Nonlinear constraints were handled via an absolute value penalty function, leading to a problem with bound constraints only. The resulting single-objective optimization problem was solved by DIRECT, an efficient deterministic direct-search global optimization algorithm, in an attempt to obtain globally Pareto-optimal solutions. Global, non-dominated solutions, whose practical importance is often overlooked, are crucial to successful MOO, in that locally Pareto-optimal (dominated) solutions mislead the DM's conception of the Pareto front and can jeopardize the exploration process of the front itself.

The proposed method was tested on a real simulation-based MOO problem, namely micro-geometry optimization of spiral bevel and hypoid gears. Such a design problem is characterized by considerable nonlinearities (hence multiple local minima) and simulation-induced nonsmoothness. Two numerical test cases were presented: In both cases, the method produced globally Pareto-optimal (and feasible) solutions, well spread over the Pareto front. The optimization required a very reasonable number of function evaluations.

All theoretical aspects, relevant references, and algorithmic details have been fully provided, along with practical considerations and technicalities, with the intention of enabling interested readers, particularly gear designers, to implement or adapt the proposed method to their design optimization purposes

No mention was made of parallelization, but it can significantly speed up the solution process. It should be pointed out that at least two levels of parallelization are possible here: (i), parallel evaluations of the objective function are available in DIRECT, and (ii), exploration of the Pareto front by the reference point method lends itself well to parallelization.

The proposed framework, albeit developed with gear optimization in mind, is obviously broader in scope and should be put to test in other computer-assisted engineering optimization problems.

References

[1] D. Ghribi, J. Bruyère, P. Velex, M. Octrue, M. Haddar, Multi-objective optimization of gear tooth profile modifications, in: Design and Modeling of Mechanical Systems, Springer, 2013, pp. 189–197.

[2] H. Ding, J. Tang, Y. Zhou, J. Zhong, G. Wan, A multi-objective correction of machine settings considering loaded tooth contact performance in spiral bevel gears by nonlinear interval number optimization, Mechanism and Machine Theory 113 (2017) 85–108.

[3] V. Simon, Optimal machine-tool settings for the manufacture of face-hobbed spiral bevel gears, Journal of Mechanical Design 136 (2014) 081004.

[4] K. Deb, S. Jain, Multi-speed gearbox design using multi-objective evolutionary algorithms, Journal of Mechanical Design 125 (2003) 609–619.

[5] K. M. Miettinen, Nonlinear Multiobjective Optimization, Kluwer Academic Publishers, Norwell, MA, USA, 1999.

[6] K. Deb, Multi-objective Optimization Using Evolutionary Algorithms, John Wiley & Sons, Chichester, West Sussex, England, 2001.

[7] J. Branke, K. Deb, K. Miettinen, R. Słowiński (Eds.), Multiobjective Optimization – Interactive and Evolutionary Approaches, volume 5252 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin Heidelberg, Germany, 2008.

[8] K. Deb, S. Agarwal, A. Pratap, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE transactions on Evolutionary Computation 6 (2002) 182–197.

[9] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the Strength Pareto Evolutionary Algorithm for multiobjective optimization, in: K. C. Giannakoglou (Ed.), Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001), International Center for Numerical Methods in Engineering (CIMNE), 2002, pp. 95–100.

[10] G.-Z. Fu, H.-Z. Huang, Y.-F. Li, B. Zheng, T. Jin, Multi-objective design optimization for a two-stage transmission system under heavy load condition, Mechanism and Machine Theory 122 (2018) 308–325.

[11] D. Miler, D. Žeželj, A. Lončar, K. Vučković, Multi-objective spur gear pair optimization focused on volume and efficiency, Mechanism and Machine Theory 125 (2018) 185–195.

[12] J. Wang, S. Luo, D. Su, Multi-objective optimal design of cycloid speed reducer based on genetic algorithm, Mechanism and Machine Theory 102 (2016) 135–148.

[13] D. R. Jones, A taxonomy of global optimization methods based on response surfaces, Journal of Global Optimization 21 (2001) 345–383.

[14] J. D. Martin, T. W. Simpson, On the use of kriging models to approximate deterministic computer models, in: Proceedings of DETC'04: ASME 2004 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, ASME, Salt Lake City, UT, USA.

[15] G. G. Wang, S. Shan, Review of metamodeling techniques in support of engineering design optimization, Journal of Mechanical Design 129 (2007) 370–380.

[16] J. A. Korta, D. Mundo, Multi-objective micro-geometry optimization of gear tooth supported by response surface methodology, Mechanism and Machine Theory 109 (2017) 278–295.

[17] A. Artoni, M. Gabiccini, M. Guiggiani, A. Kahraman, Multi-objective ease-off optimization of hypoid gears for their efficiency, noise, and durability performances, Journal of Mechanical Design 133 (2011) 121007.

[18] A. Artoni, M. Guiggiani, A. Kahraman, J. Harianto, Robust optimization of cylindrical gear tooth surface modifications within ranges of torque and misalignments, Journal of Mechanical Design 135 (2013) 121005.

[19] Y. Sawaragi, H. Nakayama, T. Tanino, Theory of Multiobjective Optimization, Academic Press, Orlando, FL, USA, 1985.

[20] A. P. Wierzbicki, The use of reference objectives in multiobjective optimization – theoretical implications and practical experience, WP-79-66, International Institute for Applied Systems Analysis (1979). Laxenburg, Austria.

[21] P. E. Gill, W. Murray, M. H. Wright, Practical Optimization, Academic Press, London and New York, 1982.

[22] J. Nocedal, S. J. Wright, Numerical Optimization, Springer Series in Operations Research and Financial Engineering, Springer, New York, NY, USA, 2006.

[23] T. G. Kolda, R. M. Lewis, V. Torczon, Optimization by direct search: New perspectives on some classical and modern methods, SIAM Review 45 (2003) 385–482.

[24] L. M. Rios, N. V. Sahinidis, Derivative-free optimization: A review of algorithms and comparison of software implementations, Technical Report, Carnegie Mellon University, Pittsburgh, PA, USA, 2010.

[25] C. A. Floudas, C. E. Gounaris, A review of recent advances in global optimization, Journal of Global Optimization 45 (2009) 3–38.

[26] K. R. Fowler, J. P. Reese, C. E. Kees, J. E. D. Jr., C. T. Kelley, C. T. Miller, C. Audet, A. J. Booker, G. Couture, R. W. Darwin, M. W. Farthing, D. E. Finkel, J. M. Gablonsky, G. Gray, T. G. Kolda, Comparison of derivative-free optimization methods for groundwater supply and hydraulic capture community problems, Advances in Water Resources 31 (2008) 743–757.

[27] D. R. Jones, C. C. Perttunen, B. E. Stuckman, Lipschitzian optimization without the lipschitz constant, Journal of Optimization Theory and Applications 79 (1993) 157–181.

[28] J. M. Gablonsky, C. T. Kelley, A locally-biased form of the DIRECT algorithm, Journal of Global Optimization 21 (2001) 27–37.

[29] W. Huyer, A. Neumaier, Global optimization by multilevel coordinate search, Journal of Global Optimization 14 (1999) 331–355.

[30] D. R. Jones, Direct global optimization algorithm, in: C. A. Floudas, P. M. Pardalos (Eds.), Encyclopedia of Optimization, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2001, pp. 431–440.

[31] J. M. Gablonsky, DIRECT Version 2.0 – User Guide, Technical Report, North Carolina State University, Raleigh, NC, USA, 2001.

[32] M. Kolivand, A. Kahraman, A load distribution model for hypoid gears using ease-off topography and shell theory, Mechanism and Machine Theory 44 (2009) 1848–1865.

[33] M. Kolivand, S. Li, A. Kahraman, Prediction of mechanical gear mesh efficiency of hypoid gear pairs, Mechanism and Machine Theory 45 (2010) 1568–1582.

[34] A. Artoni, M. Gabiccini, M. Guiggiani, Nonlinear identification of machine settings for flank form modifications in hypoid gears, Journal of Mechanical Design 130 (2008) 1126021–1126028.

[35] ANSI/AGMA 2005–D03, Design Manual for Bevel Gears, American Gear Manufacturers Association, 500 Montgomery Street, Suite 350, Alexandria, VA, USA, 2003.