

# Regularization-free multicriteria optimization of polymer viscoelasticity model

Francisco José Monaco<sup>a,\*</sup>, Roman Denysiuk<sup>b</sup>, Alexandre Claudio Botazzo Delbem<sup>a</sup>,  
Antônio Gaspar-Cunha<sup>b</sup>

<sup>a</sup> Department of Computer Systems, Institute of Mathematics and Computer Sciences, University of São Paulo, Av. do Trabalhador São-carlense, 400, Caixa Postal 668, CEP 13560-970, São Carlos SP, Brazil

<sup>b</sup> Department of Polymer Engineering, IPC Institute of Polymers and Composites, University of Minho, Campus de Azurém, 4800-058 Guimarães, Portugal

## ARTICLE INFO

### Article history:

Received 16 March 2021

Received in revised form 10 May 2022

Accepted 12 May 2022

Available online 24 May 2022

### Keywords:

Multiobjective optimization

Evolutionary computation

Polymer rheology

Computational modeling

## ABSTRACT

This paper introduces a multiobjective optimization (MOP) method for nonlinear regression analysis which is capable of simultaneously minimizing the model order and estimating parameter values without the need of exogenous regularization constraints. The method is introduced through a case study in polymer rheology modeling. Prevailing approaches in this field tackle conflicting optimization goals as a monoobjective problem by aggregating individual regression errors on each dependent variable into a single weighted scalarization function. In addition, their supporting deterministic numerical methods often rely on assumptions which are extrinsic to the problem, such as regularization constants and restrictions on parameter distribution, thereby introducing methodology inherent biases into the model. Our proposed non-deterministic MOP strategy, on the other hand, aims at finding the Pareto-front of all optimal solutions with respect not only to individual regression errors, but also to the number of parameters needed to fit the data, automatically reducing the model order. The evolutionary computation approach does not require arbitrary constraints on objective weights, regularization parameters or other exogenous assumptions to handle the ill-posed inverse problem. The article discusses the method rationale, implementation, simulation experiments, and comparison with other methods, with experimental evidences that it can outperform state-of-art techniques. While the discussion focuses on the study case, the introduced method is general and immediately applicable to other problem domains.

© 2022 Elsevier B.V. All rights reserved.

## 1. Introduction

In physical sciences and engineering, an *inverse problem* denote the goal of formulating a causal description of a given phenomenon from a set of observational experimental output data—what, mathematically, corresponds to formulating a model capable of prediction the value of one or more dependent variables based on the values of one or more independent variables. Inverse problems lay in heart of system identification field and play a role in uncountable application domains such as physics, chemistry, signal processing, machine learning, to name a few. One recognized challenging aspects of the inverse problem is that its real-world instances are often *ill-posed*, that is, they belong to the class of problems which does not satisfy Hadamard's well-posedness properties that (a) a solution exists, (b) is unique, and

(c) its behavior changes continuously with the initial conditions. An ill-posed problem may therefore have either none or many solutions; and its solutions, if any, are very sensitive to the choice of initial conditions. As consequence, stable algorithms are less likely to be applicable to solve it, and exogenous assumptions may be required such as by means of regularization or constraints in parameter value distribution. For instance, *a priori* premises on the order of the model, requirement of preset evenly spaced parameters, and objective space smoothening are widely used to tackle ill problems. Adding information which are extrinsic to the data, however, are prone to yield biased models, which are not necessarily optimal with respect to other potential solutions residing in the entire feasible solution-space.

Regression analysis is an important framework for solving inverse problems and a good deal of methods have been proposed for various specific problem domains. Particularly, nonlinear regression analysis accounts for an extensive research field in natural sciences and industrial branches, whose phenomena of interest are often described by more than one dependent variable.

\* Corresponding author.

E-mail addresses: [monaco@icmc.usp.br](mailto:monaco@icmc.usp.br) (F.J. Monaco),

[Roman.DENYSIUK@cea.fr](mailto:Roman.DENYSIUK@cea.fr) (R. Denysiuk), [acbd@icmc.usp.br](mailto:acbd@icmc.usp.br) (A.C.B. Delbem), [agc@dep.uminho.pt](mailto:agc@dep.uminho.pt) (A. Gaspar-Cunha).

One main issue is that minimizing regression errors of each of these variables often turn out as conflicting objectives. When this is the case, one practical way to address the challenge is by approaching it as a monoobjective problem, by aggregating all regression errors into a single scalarization function, e.g. a linear combination. Assigning weights to this objective function, however, introduces more bias into the model, and implies in an *a priori* decision on which individual objectives are prioritized.

This paper introduces a multiobjective optimization (MOP) method for nonlinear regression analysis aimed at simultaneously minimizing the model order and estimating parameter values without the need of exogenous assumptions. Our proposed non-deterministic MOP strategy aims at finding the Pareto-front of all optimal solutions with respect not only to individual regression errors, but also to the number of parameters needed to fit the data, allowing for an adaptive reduction of the model order. The approach does not require arbitrary constraints on objective weights, regularization parameters or other exogenous assumptions to handle the ill-posed inverse problem. The key method's rationales are introduced in the context of an industrial case study, which illustrate the problem and the strategy toward its solution. The sections which follow discuss the method's rationales, implementation, simulation experiments, and comparison with other methods, with experimental evidences that it can outperform state-of-art techniques. While focused on the study case, the method is general and immediately applicable to other problem domains.

### 1.1. Polymer rheology modeling

With major importance in industry, the viscoelastic properties of polymers have been extensively studied over the last decades, and a great deal of scientific research work has been devoted to the development of analytical models capable of describing such phenomenon. The time-dependent relation between stresses and strains in the material is usually modeled using Boltzmann superposition principle by a constitutive equation as

$$\tau(t) = \int_{-\infty}^t G(t-t')\gamma(t')dt' \tag{1}$$

which correlates the stress tensor  $\tau$  with the linear relaxation modulus  $G(t)$  and the rate of deformation tensor  $\gamma$  [1,2]. The relaxation modulus, in turn, is often analytically represented by the Maxwell generalized model, as a sum of exponential terms

$$G(t-t') = \sum_{i=1}^n G_i e^{-(t-t')/\lambda_i} \tag{2}$$

in which  $\lambda_i$  and  $G_i$  comprise the model parameters corresponding, respectively, to the relaxation times and their corresponding weights. The whole set of  $n$  pairs  $(G_i, \lambda_i)$  is known as the discrete relaxation spectrum of the material. In a common scenario, those model parameters are to be empirically determined from experimental data. Small-amplitude oscillatory shear [3] (SAOS), which lays among the prevailing methods aimed at this purpose, consists in applying a sinusoidal shear strain to a sample of the investigated material and measuring the phase difference between this excitation and the strain caused in the material at a particular frequency  $\omega_j$ . This frequency-domain experimental method produces as outcome the material's linear viscoelastic properties measured as the storage modulus  $G'(\omega_j)$  and loss modulus  $G''(\omega_j)$ , which maps into the time domain as

$$\tau(t) = \gamma [G'(\omega) \sin(\omega t) + G''(\omega) \cos(\omega t)] \tag{3}$$

One may then express both the storage and the loss moduli as

$$G'(\omega) = \sum_{i=1}^n G_i \frac{(\omega_i \lambda_i)^2}{1 + (\omega_i \lambda_i)^2} \tag{4}$$

$$G''(\omega) = \sum_{i=1}^n G_i \frac{(\omega_i \lambda_i)}{1 + (\omega_i \lambda_i)^2} \tag{5}$$

In practice, the experimental procedure consists in varying the frequency  $\omega_j$  along a discrete set of  $n$  values, and measuring the corresponding values of  $\tilde{G}'(\omega_j)$  and  $\tilde{G}''(\omega_j)$ . The modeling challenge may then be addressed as a nonlinear regression problem in which a set of  $n$  pairs  $(G_i, \lambda_i)$  is sought such that the theoretical  $G'(\omega_j)$  and  $G''(\omega_j)$  of Eqs. (4) and (5) best fit the empirically measured<sup>1</sup>  $\tilde{G}'(\omega_j)$  and  $\tilde{G}''(\omega_j)$ . The so-determined solution offers thus an estimated set of parameters for Eq. (2). The meaning of "best fit", in the former sentence may be quantified by any metric for the regression residuals such the customary mean square error, or other convenient choice.

### 1.2. State-of-art review

Mathematically, the determination of the parameters which fit Eqs. (4) and (5) is an instance of the aforementioned inverse problem. Not only a notably nontrivial theoretical challenge in itself, the special case of Eq. (2) has long been recognized as ill-conditioned [4,5], meaning that the model is very sensitive to either any method's bias or noise in experimental instrumentation. Even minor variations in each of those may yield major effects on regression parameters estimation, potentially reflecting high regression errors and spectra with unrealistic features [3,5].

To handle ill-posedness, auxiliary strategies may be evoked. Building upon Laun [6] formulations, Orbey and Dealy [3] discuss a linear-regression method which requires the number of parameters  $n$  to be preset with relaxation times  $\lambda_i, i = 1, 2, \dots, n$  evenly spaced along a logarithmic scale. The authors compare the approach with a variant by Honerkamp and Weese [5], which introduces the Tikhonov regularization as an additional constraint. Also discussed is a third alternative based on non-linear regression by Baumgaertel and Winter [7], who comment on the benefit of Tikhonov method to improve the solution convergence. Regularization is also used by Ramkumar et al. [8] in their approach based on a quadratic programming method capable of reducing regression error and increasing curve smoothness.

A novel technique referred to as edge-preserving regularization was proposed by Roths et al. [9] in order to overcome the limitations of common regularization methods in coping with spectra edges and large curvatures. Gerlach and Matzenmiller [10] compare two different numerical methods for determination of relaxation spectra, one built on gradient-based optimization algorithm and another combining Tschebyscheff approximation and Wolfe quadratic optimization. Both approaches require appropriate starting values for the parameters in order to converge to stable minima. Alternatively, the windowing method by Emri and Tschoegl [11] is a computational approach also built on the assumption of preset evenly distributed spectrum lines on a logarithmic scale.

More recent works have investigated different approaches for the determination of polymer relaxation spectra. Arguing about the practicality and availability of experimental equipment and procedures, Zatloukal et al. [12] proposed using capillary steady shear data in place of SOAS. As some aforementioned works, their analytical framework relies on non-linear regression and constraints on parameter distribution. Guzmán et al. [13] introduced an approach which does not uses regularization and is based on double-reptation model as an alternative to the conventional inversion of the integral equations, relying only on an *a priori* knowledge of an  $\alpha$  parameter and dynamic moduli data.

<sup>1</sup> The symbol  $\tilde{u}$  is used to denote empirically measured values of variable  $u$  throughout this text.

Nonetheless, and in spite of some reported interest in the use of continuous relaxation spectrum to describe polymer viscoelastic properties [14–16], the discrete formulation stands as the prevailing approach in the rheology framework, with small-amplitude oscillatory shear remaining an important experimental method both in scientific research and industry shop floor.

### 1.3. Evolutionary computation approach

To handle the nonlinearity and ill-posedness of the regression problem, the aforementioned and other analytical and computational deterministic methods described in the literature are, to a broad extent, reliant on some type of assumptions that are extrinsic to the experimental data, such as heuristically tuned regularization parameters, constraints on the number of parameters and their distribution. Meanwhile, in a distinctively different vein, a non-deterministic optimization method has been proposed [17] which does not rely either on arbitrary or exogenous regularizing parameters nor technique-biased parameter distribution hypotheses. The introduced method, grounded on evolutionary computation theory, was successfully applied to the determination of the discrete linear viscoelastic relaxation spectrum from SOAS data, and experimental evidences indicate that it was able to fit the storage modulus  $G'$  and the loss modulus  $G''$  of Eqs. (4) and (5) with significantly low regression error and requiring a low number of parameters  $(G_i, \lambda_i)$  relatively to other reported methods.

Essentially, an evolutionary algorithm (EA) works not too differently from other iterative optimization methods in which a solution is successively improved at each step. Unlike conventional deterministic techniques, nevertheless, the EA approach evolves not a single solution, but a collection of many prospective solutions simultaneously. At each step, the algorithm selects a few from among the best solutions found so far, and, from these, it derives some other new solutions. Both these procedures, selection and derivation, embody a degree of randomness. The best solutions have a high probability, rather than certainty, of being picked up—such a nongreed strategy is key to avoiding an otherwise inevitable premature convergence to local optima. The procedure to derive new solutions from the selected ones is performed by combining the latter, such that *offspring solutions* are close in the solution space to their *parent solutions*, already a product of best-fit selection, and hopefully may thus further approach an optimum in the objective space. Often, after recombination, mutations – minor random changes with low probability – are applied in order to generate new solutions which would not be produced solely by recombination—and thus, allowing for further exploration of the search space. Summarizing, and resorting to the metaphor of the natural evolution upon which the computational strategy is built, the successive choice of the “best fit” – evaluated by the objective function – and their recombination into similar solutions yield a selective pressure driving the convergence to optima in the solution space; probabilistic selection and random variations impose a contrary pressure which tends to avoid premature convergence and better exploration of the solution search space.

In their paper Monaco et al. [17], authors describe the realization of this approach in the form of a conventional *genetic algorithm* [18] (GA) modified to work with real-valued genes, each a 2-tuple  $(G_i, \lambda_i)$ . The method was implemented as a computer program which takes experimentally measured  $G'$  and  $G''$  data and produces as output a set of  $n$  parameter  $(G_i, \lambda_i)$  along with the computed regression errors, as in Eqs. (7) and (8). Using data sets from the literature, the evolutionary algorithm was capable of reproducing Laun [6] results for LDPE material with  $n = 8$  when abiding by the constraint of preset logarithmically

distributed  $\lambda$ . Under the same restriction, an equivalent result in the curve smoothness obtained by Ramkumar et al. [8] with  $n = 41$  parameter was achieved with a regression residuals as low as two orders of magnitude below that reported by the latter work. When the constrain on  $\lambda$  spacing was removed, the genetic algorithm reproduced Laun [6] result with a 20% lower regression residue, and that of Ramkumar et al. [8] with residuals below the measurable precision limit and equivalent smoothness. Moreover, the authors replicated this computational experiment with different choices of  $n$  and could achieve both regression residuals and curve smoothness equivalent to that of  $n = 41$  evenly *log-scale* distributed  $\lambda$  found by said work [8], but with as few as  $n = 8$  free- $\lambda$  (unconstrainedly distributed relaxation times). Results for other materials including HDPE, PB and Honerkamp and Weese [5] synthetic data were produced—in those cases, the genetic algorithm needed, respectively  $n = 4, 6$  and  $7$  parameter pairs to outperform the reported results and, respectively,  $n = 8, 10$  and  $10$  to reach the minimum regression residual, beyond which no improvement was seen by increasing  $n$ .

It is fair to remark, at this point, that whereas such a positive outcome might not be assured beforehand – bearing in mind the approach's inherent non-determinism – it is nevertheless not absolutely surprising, once taken into account that the less constraints are imposed, the broader the range or possible solutions. In this case, the problem admits superior (best fitted) choices of regression parameters out of the evenly *log-distributed*  $\lambda$  restriction. Indeed, the self-sufficiency to explore the solution search space without the need of arbitrary, extrinsic constraints is a noteworthy inherent virtue of the evolutionary computational approach. Evolutionary algorithms (of which genetic algorithms are an example) are known as a metaheuristic, meaning that it is not a specific-purpose prescription for a particular problem, but rather a generic problem-solving mechanism which can be tailored to different domains. Given a codification scheme to represent prospective solutions (chromosome), a procedure to derive new solutions from existing ones (recombination and mutation), and an algorithm to compare the quality of two solutions (fitness function), a GA is essentially a well-reasoned trial-and-error strategy, grounded on natural evolution theory and backed by empirical positive evidences of both biological and artificial systems, implemented as a non-deterministic iterative optimization algorithm.

### 1.4. The multiobjective problem

The issue motivating the present research work concerns the fact that the goal of determining the parameters that minimize the regression residues of both  $G'$  and  $G''$  theoretical curves translates into a multiobjective optimization problem (MOP). That is because the same set of parameters which minimizes the residue of Eq. (4) may not be the best choice to minimize the residue of Eq. (5). If we associate each individual regression a metric of quality which quantifies the resulting residues, we may define two objective functions  $f_1$  and  $f_2$ . As is usual in real multiobjective optimization problems, in the problem under investigation both objectives are conflicting.

One straightforward strategy to tackle the goal of minimizing  $m$  objective functions at the same time is to address it as a monoobjective problem by means of scalarization. Under this approach, if a solution  $\mathbf{x}$  is sought which minimizes  $\mathbf{f} = \{f_1(\mathbf{x}), f_2(\mathbf{x}) \dots f_m(\mathbf{x})\}$ , then a scalar function  $g(\mathbf{x}, \mathbf{w})$  is chosen as

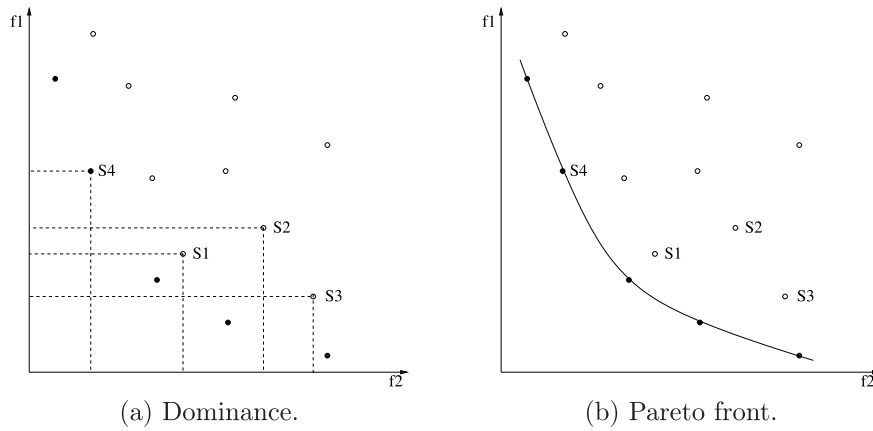


Fig. 1. Pareto set as the solution of a multiobjective optimization problem.

the new monobjective function to be optimized. For instance, the *weighted sum* method corresponds to a linear combination of the individual objective functions  $f_i$ , such as

$$g(\mathbf{x}, \mathbf{w}) = w_1 f_1(\mathbf{x}) + w_2 f_2(\mathbf{x}) + \dots + w_k f_m(\mathbf{x})$$

$$\sum_{i=1}^k w_i = 1, \quad w_i \geq 0$$

is one very simple, yet meaningful scalarization function. The set of weights  $\mathbf{w} = \{w_1, w_2, \dots, w_m\}$  may be tuned to reflect the relative importance of each objective. The exploration of the solutions' search space is narrowed and driven around the possibilities which conform to the chosen weights. This is an example of an *a priori* multiobjective problem solving technique in which the decision maker assigns optimization priorities beforehand, accounting for the strategy exploited in aforementioned work [17]. In this present paper we extend those results by tackling the same challenge as a strictly multiobjective optimization problem. That is, we aim at finding not one single best solution, but a set of solutions which are the optimal ones with respect to all possible weight assignments, allowing thus for *a posteriori* decision making on the particular solution to be chosen. This collection of optimal solutions is commonly referred to as the Pareto set. The concept may be intuitively depicted as in Fig. 1, addressing a generic bi-objective problem. In the plot, the two objective functions  $f_1$  and  $f_2$ , indicated by orthogonal axes, define the *objective space*. In this bidimensional example, each dot on the solution's plane corresponds to a particular set of parameters in the feasible solution space—which may have a different dimension.

Like in the running case study, where the objective functions are regression residuals, let us suppose that the goal is to minimize both  $f_1$  and  $f_2$ . In Fig. 1(a), solution S1 is clearly better than solution S2, seeing that the former minimizes both objectives respectively to the latter. In the optimization field's terminology, S1 is said to *dominate* S2. A like conclusion is not however so immediate when comparing S1 to S3, as either one is better respectively to each of the objectives—bringing up the fundamental challenge of defining “optimum” in a multiobjective problem. The formal approach for this dilemma is to define that a given solution is dominated if there is another solution which is better than it with respect to at least one of the objectives. Therefore, both solutions S1 and S3 are dominated. On the other hand, solution S4 is *nondominated*, inasmuch as there is no other solution which is better than it regarding any objective. In the plot of Fig. 1(a), white dots are dominated solutions, whereas black dots are nondominated ones. If over all the solution search space there are no feasible solutions that dominate the ones

shown in the plot, the nondominated solutions are considered optimal solutions, meaning that any attempt to improve them with respect to one objective would degrade another objective. The frontier on which lay the optimal solutions, depicted in Fig. 1(b), is known as the Pareto front.

In possession of the Pareto front, the problem analyst may choose among all optimal solutions, selecting which is more appropriate based on a particular objective priority assignment. For instance, they can prefer a solution which equalizes all objectives, or else a solution which relatively prioritizes more critical objectives to the detriment of less important ones. In the focused problem, the domain analyst may make decisions on the basis of accuracy or precision associated with either  $G'$  or  $G''$  data sets, or any other sensible criteria.

## 2. Problem formulation

To comply with an usual preference in the literature, we herein quantify the regression quality as the mean absolute percentage error<sup>2</sup>  $\epsilon$ , as hereafter denoted. Formally, for a set of  $d$  data values  $x_i$ , this metric is defined as

$$\epsilon = \frac{1}{d} \sum_{j=1}^d \left| \frac{\tilde{x} - x}{\tilde{x}} \right| \tag{6}$$

with  $\tilde{x}$  indicating the experimental data and  $x$ , the value forecast by the model.<sup>3</sup> Formally, thus, the addressed multiobjective problem may thus be stated as the goal of minimizing the objective-functions  $\epsilon_1$  and  $\epsilon_2$  given by the expressions (7) and (8) as

$$\epsilon_1(P) = \frac{1}{d'} \sum_{j=1}^{d'} \left| \frac{\tilde{G}'(\omega_j) - G'(\omega_j)}{\tilde{G}'(\omega_j)} \right| \tag{7}$$

$$\epsilon_2(P) = \frac{1}{d''} \sum_{j=1}^{d''} \left| \frac{\tilde{G}''(\omega_j) - G''(\omega_j)}{\tilde{G}''(\omega_j)} \right| \tag{8}$$

<sup>2</sup> In evaluating a prediction model, the mean absolute percentage error  $\epsilon$  between a series of forecast values  $f_i$  and their corresponding actual values  $a_i$ , is a measure of the absolute relative deviation between the actual and forecast values. i.e. the average of  $|f_i - a_i|$  over the  $n$  data records.

<sup>3</sup> Some authors, e.g. [3,8] use to refer to this metric as the *average absolute deviation* (AAD). This however conflicts with the conventional statistics terminology, which defines AAD as the average deviation of each individual variable with respect to the sample's mean—therefore, as a dispersion rather a regression quality measurement. In this paper we abide with the more precise denomination of *mean absolute percentage error*, as commonly used in statistics field.

- 
- a. Generate an initial population with  $\rho$  random individuals (solutions)
  - b. Evaluate the fitness (objective function) of each individual
  - c. Select a subset of the population with  $\beta \leq \rho$  best individuals
  - d. Derive  $\beta$  new individuals by combining the ones selected in step (c)
  - e. Replace up to  $\beta$  individuals of the population with the best newly created ones
  - f. Repeat from step (b) until a stop criterion is reached
- 

**Fig. 2.** Outline of a monobjective GA.

$$P = \{p_1, p_2, \dots, p_n\}, \quad p_i = (G_i, \lambda_i)$$

$$G_i \in [G_{min}, G_{max}], \quad \lambda_i \in [\lambda_{min}, \lambda_{max}] \quad (9)$$

with  $P$  being the set of  $n$  determined parameter  $p_i = (G_i, \lambda_i)$ ,  $i = 1, 2, \dots, n$ . In the equation,  $G'$  and  $G''$  stand for the storage and loss moduli;  $d'$  and  $d''$  account for the number of available experimental data, respectively. The tilde symbol indicates the experimental values and the corresponding tilde-less variables hold the values forecast by the model. Condition (9) delimits the feasible solution space which the algorithm is allowed to explore.

In this paper, we exploit a multiobjective evolutionary algorithm to generate a Pareto set containing optimal solutions for the determination of the linear viscoelastic relaxation spectrum from experimental data. Each solution is a set of parameters  $(G_i, \lambda_i)$  which fits the theoretical curve of the storage and loss module given by Eqs. (4) and (5), and is optimal in the sense that it yields the best possible  $\epsilon_1$  for an admitted  $\epsilon_2$  and vice versa. Moreover, the method allows for including the number of parameters  $n$  as one of the multiple objectives to be optimized, thereby offering the decision maker to appreciate the trade-offs between regression quality and model order. Low-order models are advantageous for reducing cost and processing time of computational-intensive simulations.

### 3. The algorithm MOEA/D

When tackling the running case study by means of an evolutionary strategy, Monaco et al. [17] outlined the working principle of a monobjective GA in the form of a simple algorithm, adapted in Fig. 2.

Under the metaphor of genetic algorithms, the initial population of step (a) is a randomly generated set of  $\rho$  prospective solutions encoded in an appropriate form. In the running study case, an individual is a candidate solution of the regression problem encoded as an ordered sequence of  $n$  pairs  $(G_i, \lambda_i)$ , which plays the role of a chromosome. In step (b), the fitness function is the regression quality, which the authors assumed as the simple average between  $\epsilon_1$  and  $\epsilon_2$ , given by Eqs. (7) and (8). In step (c), a “mating pool” is formed by the probabilistic selection of  $\beta$  “best fitted” individuals, that is, those chromosomes (parameter sets) yielding lower average regressing residuals. In step (d), each of the  $\beta$  new individuals is formed such that some have their genes –  $(G_i, \lambda_i)$  pairs – closer to a “father” solution, whereas others have their gene closer to a “mother” solution among the ones in the mating pool, in a vague analogy to a biological offspring. Finally, in step (e) the new individuals are compared with all individuals of the original population regarding fitness; the highly scored ones survive to the detriment of the worst fitted, again probabilistically. The process repeats until a stop criterion is reached, which, in the present example, is when the average regression error eventually ceases to show significant improvement at each new iteration. The GA output is a sequence of  $n$  pairs  $(G_i, \lambda_i)$  which best fit Eqs. (4) and (5). It is worthy to remark that, as a non-deterministic numerical method, the solution is not guaranteedly the absolute optimal, but a best-effort estimation. Successive runs of the algorithm with varying initial conditions

and random selection may yield numerically distinct results. The reported computational method was however evaluated as robust in the sense that the generated solutions, yet different, result in equivalent regression quality.

Contrasting to this algorithm, a multiobjective evolutionary algorithm (MOEA) builds upon a different paradigm as explained in Section 1.4. Even so, some general principles of evolutionary computational methods are shared among the mono and multiobjective algorithms. Both start with an initial population and iteratively evolve it through successive steps of random selection of the best fitted, random recombination and variation, with parameters set to boost selective pressure (convergence toward the optimum) and diversity preservation (solution space exploration). The main difference between a monobjective GA and a MOEA is that, while the former (GA) aims at driving a unique individual towards a single optimum – possibly a scalar combination of several objectives – the latter (MOEA) aims at driving the whole population to settle along the Pareto front, considering all objectives independently.

Perhaps the most straightforward strategy to achieve this is to start from the very concept of dominance, and design a selection policy which probabilistically preserves the nondominated solutions to the detriment of the dominated ones. Its not difficult to see that this will tend to cause the surviving solutions to approach the Pareto front. One problem with this approach, which has been long noticed [19], is that this sole criterion does not prevent the tendency of all found Pareto-optimal solutions to cluster together in one region of the theoretical Pareto front, due to the inherent convergence pressure of the evolutionary process. Aiming at a more even distribution of solutions along the Pareto front, several techniques have been proposed [20], many of which embedding into the selection mechanism some rule to prefer scattered over crowded solutions. A particularly ingenious one, suggested by Zitzler and Thiele [21], is using to use a metric called hypervolume, graphically explained in Fig. 3. The hypervolume is the union of the areas of all rectangles with one vertex at a reference point  $Z$  and the opposite vertex at a candidate optimal solution. It is easy to see that the more evenly spread the set of Pareto solutions, the larger the resulting area. In the image,  $Z$  is arbitrary point that is at least slightly worse than the nadir point of the Pareto front approximation obtained.

While effective, all the proposed techniques, including the hypervolume, are computationally costly, especially in higher dimension problems, and with many objectives. Because of this, a different approach suggested by Zhang and Li [22] proposes to tackle the problem by means of a decomposition strategy.

Decomposition is well-known in multiobjective optimization. Essentially, it consists in converting the MOP into a set of monobjective problems through some scalarization function. For instance, using the weighted sum method, a two-dimensional MOP with objectives  $f_1(\mathbf{x})$  and  $f_2(\mathbf{x})$  may be converted into a scalar optimization problem whose goal is to optimize (either maximize or minimize) a function  $g(\mathbf{x}, \mathbf{w}) = w_1 f_1(\mathbf{x}) + w_2 f_2(\mathbf{x})$ , with  $w_1 + w_2 = 1$  and  $w_i \in [0, 1]$ . The vector  $\mathbf{w} = (w_1, w_2)$  is

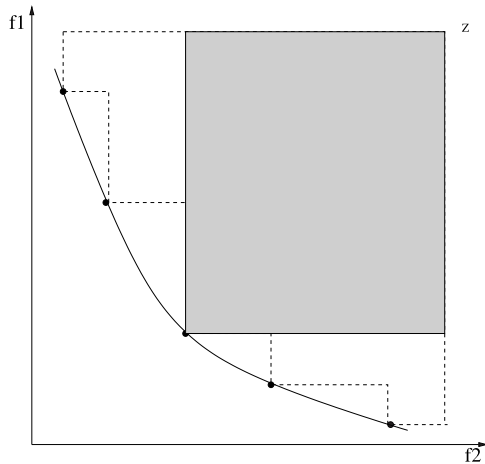


Fig. 3. Hypervolume index.

referred to as the *weight vector*, while  $\mathbf{f} = (f_1, f_2)$  is the *objective-function vector*.<sup>4</sup> The optimal solution of the scalar function  $g$ , with a particular weight vector  $\mathbf{w}$  is a point in the Pareto front of the MOP problem. Therefore, by taking  $k$  weight vectors  $\mathbf{w}^j \in w^1, w^2, \dots, w^k$  and optimizing each  $g(\mathbf{f}, \mathbf{w}^j)$  individually, one can obtain  $k$  Pareto-optimal solutions of the MOP. Generalizing the preceding reasoning in a concise formulation, the decomposed MOP problem may therefore be stated as follows.

$$\mathbf{f}(\mathbf{x}) = \{f_i(\mathbf{x})\} \quad i = 1, 2, \dots, m, \quad \mathbf{x} \in \Omega \quad (10)$$

$$W = \{\mathbf{w}^j\} \quad j = 1, 2 \dots k \quad (11)$$

$$\mathbf{w}^j = \{w_i^j\} \quad i = 1, 2, \dots, m, \quad \sum_{i=1}^m w_i^j = 1, w_i^j \geq 0 \quad (12)$$

$$S(\mathbf{f}^j(\mathbf{x}), \mathbf{w}^j, \dots) = g(\mathbf{x}, \mathbf{w}^j, \dots) \quad (13)$$

In Eq. (10), let  $\mathbf{f}(\mathbf{x})$  be a set of  $m$  objective-functions whose domain  $\Omega$  is the *feasible solution space*, i.e. where lay the solutions of practical interest. The objective-functions' argument  $\mathbf{x}$  may be multidimensional and comprise the set of *decision variables* concerning the problem. The MOP decomposition implies addressing the aggregation of objectives according to relative weights, and searching for the Pareto-optimal set requires several weight vectors. Let  $W$ , in Eq. (11), be this set of  $k$  weight vectors  $\mathbf{w}^j$ , each of these a vector itself with  $m$  non-negative coordinates  $w_i^j$  which sum up 1, as in Eq. (12). As for the scalarization method, we may define a generic operator  $S$ , as in Eq. (13) which associates both the objective and a particular weight vectors to a scalar function  $g(\mathbf{x}, \mathbf{w}^j, \dots)$ —the ellipsis is left as a provision for scalarization functions depending on other parameters. For instance, the weighted sum is an example of a scalar operator  $S_{ws}(\mathbf{f}(\mathbf{x}), \mathbf{w}^j) = g_{ws}(\mathbf{x}, \mathbf{w}^j) = w_1^j f_1(\mathbf{x}) + w_2^j f_2(\mathbf{x}) + \dots + w_m^j f_m(\mathbf{x})$ . Solving the MOP/D (multiobjective problem with decomposition) system involves optimizing  $g$  for all  $\mathbf{w}^j, j = 1, 2 \dots k$  so as to obtain a well-distributed set of solutions as close as possible to the theoretical Pareto front.

While determining the  $k$  optimal solution in this way, one each a time, may be a simple and effective approach, this is however not necessarily a very efficient one. Other methods have been proposed [20] to optimize all scalar functions simultaneously at each iteration. One such method, which we address in the present paper, is a rather clever mechanism proposed by Zhang

and Li [22], named *multiobjective evolutionary algorithm based on decomposition*, MOEA/D. In general terms, the key idea of MOEA/D is to choose evenly distributed weight vectors and exploit the observation that, for two close weight vectors, the optimal solutions of their corresponding scalar functions tend to be close together along the Pareto front—seeing that  $g^j$  is linear to  $w^j$ . MOEA/D takes advantage of this property by allowing recombination between evolving solutions of neighbor scalar functions, as measured by the Euclidean distance between weight vectors. That is, at each iteration of the algorithm, the evolution of each monobjective problem benefits from the progress of the neighbor problems, and as result, the whole set of solutions converge to the Pareto front faster and in a well-distributed manner. The technical details and in-depth analysis of the MOEA/D algorithm are offered in the proponents' article [22].

#### 4. MOEA/D implementation

In the present research work, we have designed the Algorithm 1 for the MOEA/D method.

The algorithm declares (line 2)  $W$  an ordered set of  $k$  weight vectors  $\mathbf{w}^j$ . Then, for each  $\mathbf{w}^j$ , it defines a neighborhood with  $s \leq k$  other vectors which, further on, will be optimized in combination as outlined in Section 3. In the algorithm, (line 3)  $B^j$  is the set of indexes corresponding to the  $\mathbf{w}^j$ 's neighbor vectors. Finally,  $P$  is the population (line 4) of sought after solutions  $\mathbf{x}^j$ . The procedure starts by initializing  $W$  (line 6) with  $k$  evenly distributed weight vectors<sup>5</sup> satisfying the conditions of Eq. (12). The neighborhood of each  $\mathbf{w}^j$  is initialized (line 7) with the  $s$  nearest weight vectors, as given by their Euclidean distances, while the population  $P$  is initialized (line 8) with randomly generated solutions (solution vectors)  $\mathbf{x}_i$  within the allowable bounds of the feasible solution space  $\Omega$ . The population is evolved across  $t_{max}$  generations (iterations), as controlled by the main program loop (line 9). Within it, each solution  $\mathbf{x}^j$  is visited and improved as follows. First, its neighbor solutions (corresponding to each scalar problem, and each weight vector) are combined (lined 12)—the exact variation operator  $\mathcal{V}(B^j, W)$  by which this combination is performed is left open for flexibility. Then, the newly generated solution  $\mathbf{y}$  produced at this step is compared to its neighbors (line 15) by means of their corresponding scalar functions  $g$ —again, the scalar operator  $S(\mathbf{f}^j(\mathbf{x}), \mathbf{w}^j, \dots)$  to derive  $g(\mathbf{x}, \mathbf{w}^j)$  is left open. This comparison is the basis to decide on which solutions are replaced and which are retained (line 16). Parameter  $\tau_{max}$  controls how many current solutions may be so replaced. Algorithm 1 was implemented using MATLAB<sup>6</sup> framework.

##### 4.1. Choices and parameters controlling the algorithm

As mentioned, the choices of both the scalar and variation operators allow for flexibility and exploration through empirical investigation, as addressed in the following sections.

Another central aspect of an evolutionary algorithm implementation is the solution encoding scheme. This choice not only impacts the performance but also guides the design of the variation operators. In our implementation, we have opted for a quite simple scheme in which each prospective solution (a chromosome in GA jargon) of  $n$  parameters is represented by an array of  $2 \times n$  real values, with even positions corresponding to values of  $\log(G_i)$  and odd positions to values of  $\log(\lambda_i)$ . This allows for variation operators for continuous real-valued search spaces already existing for monobjective EAs to be used out of the

<sup>4</sup> We herein use boldface to denote vector variables and functions in order to differentiate from scalar counterparts.

<sup>5</sup> Programmatically,  $\mathbf{w}^j = (j/k, (k-j)/k)$ ; refer to the source code for further implementation details.

<sup>6</sup> <https://www.mathworks.com/products/matlab.html>.

**Algorithm 1** Our MOEA/D design.

```

1: //Declarations
2:  $W = \{w^1, w^2 \dots w^k\}$ 
3:  $B = \{B^1, B^2 \dots B^k\}$ 
4:  $P = \{x^1, \dots, x^k\}$ 
5: // Initialization
6:  $W \leftarrow \text{init}_W()$ 
7:  $B \leftarrow \text{init}_B()$ 
8:  $P \leftarrow \text{init}_P()$ 
9: for  $t = 1, \dots, t_{\max}$  do
10:   for  $j = 1, \dots, k$  do
11:     // Variation
12:      $y \leftarrow \mathcal{V}(B^j, W)$ 
13:     // Replacement
14:     for  $\tau = 1, 2 \dots s$  and  $\tau \leq \tau_{\max}$  do
15:       if  $g(y, w^\tau \dots) > g(x^\tau, w^\tau, \dots)$  then
16:          $x^\tau \leftarrow y$ 
17:       end if
18:     end for
19:   end for
20: end for

```

$$w^j = (w_1^j, w_2^j \dots w_m^j), w_i^j \in \mathbb{R}$$

$$B^j = \{1, 2 \dots s\}, s \leq k$$

$$x_i^j \in \Omega$$

$$w_i^j \in [0, 1], \sum_{i=1}^m w_i^j = 1, \|w^a - w^b\|_2 = \|w^c - w^d\|_2$$

$$\forall a \in B^j, \forall b \notin B^j : \|w^j - w^a\|_2 \leq \|w^j - w^b\|_2$$

$$x_i : \text{random}(\Omega)$$

shelf, what comes in handy when comparing results with other research works.

The algorithm reads as input two text files containing the experimental data in CSV format,<sup>7</sup> one for the storage modules  $G'$  and another for the loss module  $G''$ . The procedure is also controlled by the choices of  $k$ , the number of desired solutions in the Pareto front;  $t_{\max}$ , the number of generations (iterations) across which the solution population is to be evolved (optimized); and  $s$ , the size of each solution's neighborhood considered in the offspring (derived solution) generation. All these parameters may be empirically tailored for the particular problem and data sets—they will mostly affect converge speed.

## 5. Implementation specifics

Aiming at verifying the susceptibility of the result quality with respect to the choice of the scalarization and variation operators, we have implemented a few already existing alternatives.

### 5.1. Variation operators

As for the variation operators  $\mathcal{V}(B^j, W)$ , three well-known recombination methods were implemented.

#### 5.1.1. Simulated binary crossover

A Simulated Binary Crossover (SBX) [23] attempts to simulate the working principle of a single-point crossover for binary strings. Although a crossover is typically performed between a pair of parent solutions to obtain a pair of offspring, a single offspring is produced in MOEA/D during recombination. Thus, the SBX operator can be expressed as:

$$y_j = \begin{cases} 0.5 \left( (1 + \beta_j) x_j^{(1)} + (1 - \beta_j) x_j^{(2)} \right) & \text{if } r_j \leq 0.5 \\ 0.5 \left( (1 - \beta_j) x_j^{(1)} + (1 + \beta_j) x_j^{(2)} \right) & \text{otherwise} \end{cases} \quad (14)$$

with the values of  $\beta_i$  being obtained from the following distribution:

$$\beta_j = \begin{cases} (2u_j)^{\frac{1}{\eta_c+1}} & \text{if } u_j \leq 0.5 \\ \left( \frac{1}{2(1-u_j)} \right)^{\frac{1}{\eta_c+1}} & \text{otherwise} \end{cases} \quad (15)$$

<sup>7</sup> Comma-separated value. In our implementation each row is a data measurement where the first column represents the frequency  $\omega$ , and the second column is the associated measurement.

for  $j = 1, \dots, n$ , where  $r_j, u_i \sim \mathcal{U}(0, 1)$  and  $\eta_c > 0$  is an external parameter of the distribution.

#### 5.1.2. Differential evolution

A differential evolution (DE) [24] operator generates a new solution  $y$  by exploiting the differences between solution vectors in the population and can be described by:

$$y_j = \begin{cases} x_j^{(1)} + F(x_j^{(2)} - x_j^{(3)}) & \text{if } r_j \leq CR \\ x_j^{(1)} & \text{otherwise} \end{cases} \quad (16)$$

for  $j = 1, \dots, n$ ,  $F$  is the scale factor,  $CR$  is the crossover probability and  $r_i \sim \mathcal{U}(0, 1)$ .

#### 5.1.3. Gaussian model

A multivariate Gaussian model (GM) [25] operator samples solutions from:

$$x \sim \mathcal{N}(\mu, \Sigma) \quad (17)$$

where  $\mu \in \mathbb{R}^n$  is the mean vector and  $\Sigma \in \mathbb{R}^{n \times n}$  is the covariance matrix of the probability distribution. The mean of the Gaussian model is assumed to be the solution associated with the  $i$ th subproblem:

$$\mu = x^{(i)}. \quad (18)$$

The elements of  $\Sigma$  are computed using neighboring solutions and an unbiased estimate of the covariance:

$$\Sigma_{i,j} = \frac{1}{K-1} \sum_{k=1}^K (x_i^{(k)} - \mu_i)(x_j^{(k)} - \mu_j) \quad (19)$$

for  $i, j = 1, \dots, n$ , where  $K$  is the size of the considered neighborhood.

#### 5.1.4. Polynomial mutation

In addition to the aforementioned recombination operators, a mutation operator was used in association with all three of those, as a subsequent step in the variation procedure. A Polynomial Mutation (PM) [26] is typically applied to introduce additional variations to newly generated solutions so as to ensure the exploration of new regions of the search space. This operator gives a higher probability that a new solution  $y$  is closer to rather than far away from the previous solution  $x$ . It can be expressed by:

$$y_i = \begin{cases} x_i + (u_i - l_i)\delta_i & \text{if } r_j \leq p_m \\ x_i & \text{otherwise} \end{cases} \quad (20)$$

with the values of  $\delta_i$  being given by:

$$\delta_j = \begin{cases} (2u_j)^{\frac{1}{\eta_m+1}} - 1 & \text{if } u_j \leq 0.5 \\ 1 - (2 - 2u_j)^{\frac{1}{\eta_m+1}} & \text{otherwise} \end{cases} \quad (21)$$

for  $j = 1, \dots, n$ , where  $r_j, u_i \sim \mathcal{U}(0, 1)$ , whereas  $p_m \in [0, 1]$  and  $\eta_m > 0$  are control parameters.

### 5.2. Scalarization operators

Respecting scalarization operators  $S(\mathbf{f}^j(\mathbf{x}), \mathbf{w}^j, \dots)$ , again three well-known techniques were implemented.

#### 5.2.1. Chebyshev method

Chebyshev (CHB) method belongs to the group of weighted metric methods [27] that aims at minimizing the distance between a reference point  $\mathbf{z} = (z_1, \dots, z_m)$  and the feasible objective region [27]. The CHB scalarization is given by:

$$\text{minimize } F(\mathbf{f}, \mathbf{w}) = \max_{1 \leq i \leq m} w_i |f_i - z_i|. \quad (22)$$

This method can find solutions in convex and nonconvex regions of the Pareto front. The drawbacks are that it cannot distinguish weakly Pareto optimal solutions and does not provide a uniform distribution of solutions along the Pareto front.

#### 5.2.2. Penalty boundary intersection method

The penalty boundary intersection (PBI) [28] is developed in an attempt to generate a more uniform set of Pareto optimal solutions and is given by:

$$\text{minimize } F(\mathbf{f}, \mathbf{w}) = d_1 + \theta d_2 \quad (23)$$

where

$$\begin{aligned} d_1 &= \frac{\|(\mathbf{f} - \mathbf{z})^T \mathbf{w}\|}{\|\mathbf{w}\|} \\ d_2 &= \|\mathbf{f} - (\mathbf{z} + d_1 \frac{\mathbf{w}}{\|\mathbf{w}\|})\|. \end{aligned} \quad (24)$$

The major advantage of this method is that it can provide a reasonably uniform distribution of solutions along the Pareto front. Though it comes at the cost of specifying the value of  $\theta$ .

#### 5.2.3. Weighted stress function method

The weighted stress function method (WSFM) [29] draws inspiration from the stress-strain behavior of thermoplastic vulcanizates. Using the weight  $w_i$ , the WSFM transforms the  $i$ th objective  $f_i$  into a stress  $\sigma_i$ . This requires  $f_i \in [0, 1]$  and  $w_i \in (0, 1)$ , which is achieved by:

$$f_i = \frac{f_i - f_i^{\min}}{f_i^{\max} - f_i^{\min}} \quad (25)$$

and

$$w_i = \min\{\max\{w_i, \epsilon\}, 1 - \epsilon\} \quad (26)$$

for  $i = 1, \dots, m$ , where  $f_i^{\min}$  and  $f_i^{\max}$  are the minimum and maximum values of the  $i$ th objective in the current population, whereas  $\epsilon$  is a small positive value.

The WSFM scalarization aims at minimizing the largest stress associated with the given solution:

$$\text{minimize } F(\mathbf{f}, \mathbf{w}) = \max_{1 \leq i \leq m} \sigma_i(f_i, w_i). \quad (27)$$

with being calculated as:

$$\sigma_i(f_i, w_i) = (1 + \omega_i(f_i, w_i)) \xi_i(w_i) \quad (28)$$

where

$$\omega_i(f_i, w_i) = \begin{cases} \frac{\tan\left(\frac{\pi}{\psi_i(w_i)}(f_i - w_i)\right)}{\tan\left(\frac{\pi}{\phi_i(w_i)}(w_i - \delta_1)\right)} \frac{\psi_i(w_i)}{\phi_i(w_i)}, & f_i \geq w_i \\ \frac{\tan\left(\frac{\pi}{\phi_i(w_i)}(f_i - w_i)\right)}{\tan\left(\frac{\pi}{\phi_i(w_i)}w_i\right)}, & f_i < w_i \end{cases} \quad (29)$$

$$\psi_i(w_i) = \frac{3}{4}w_i^2 + 2(1 - w_i) + \delta_1 \quad (30)$$

$$\phi_i(w_i) = \frac{3}{4}w_i^2 + 2w_i + \delta_1 \quad (31)$$

$$\xi_i(w_i) = 1 - \frac{\tan\left(\frac{\pi}{2(1+\delta_2)}(2w_i - 1)\right)}{\tan\left(\frac{\pi}{2(1+\delta_2)}\right)}. \quad (32)$$

The particular feature of this method is that it results in nonlinear lines along which the search is performed.

## 6. Experimental setup

This section details a set of simulations aimed at exploring the computational method and summarizes their results. Relevant findings and pertinent discussions are contextually highlighted along the text. All experimental framework, including algorithm implementation, experimental data, and simulation results are available from the project repository.<sup>8</sup>

### 6.1. Experimental data

The study considers two types of polymer materials: Low Density Polyethylene (LDPE) and Polystyrene (PS). LDPE is a lightweight plastic material that has excellent resistance to water, moisture, and most organic solvents and chemicals. It is widely used for packaging and as a protective coating. PS is a polymer that is produced through the polymerization of styrene in monomer status. It possess good flow properties at temperatures safely below degradation ranges, and can easily be extruded, injection molded, or compression molded. Considerable quantities of polystyrene are produced in the form of heat-expandable beads containing a suitable blowing agent which ultimately results in familiar foamed polystyrene articles. The number of experimental data available for LDPE and PS are 57 and 109 measurements, respectively.<sup>9</sup> For the available data, Fig. 4 displays the plots of the storage modulus  $G'(\omega)$  and the loss modulus  $G''(\omega)$  against the frequency  $\omega$ .

### 6.2. Parameters and procedures

MOEA/D is a stochastic search algorithm. Due to this fact, the computational experiment involves multiple executions of the algorithm with different random number sequences,<sup>10</sup> and the obtained results are to be analyzed with respect to a statistical framework.

For each experiment instance, 25 independent runs were performed with different random sequences, and individual results

<sup>8</sup> Project repository: <https://gitlab.uspdigital.usp.br/research/moadpolymer>.

<sup>9</sup> The experimental data was obtained ....

<sup>10</sup> To produce random sequences out of deterministic computational procedures, programs usually rely on pseudo-random functions which compute different results upon successive calls, depending on the current value held in an internal state-variable that is automatically updated at every invocation. The so generated sequences shall fulfill general mathematical proprieties of randomness. Distinct random sequences can be produced by initializing the algorithm internal state with an initial value often referred to as random seed.



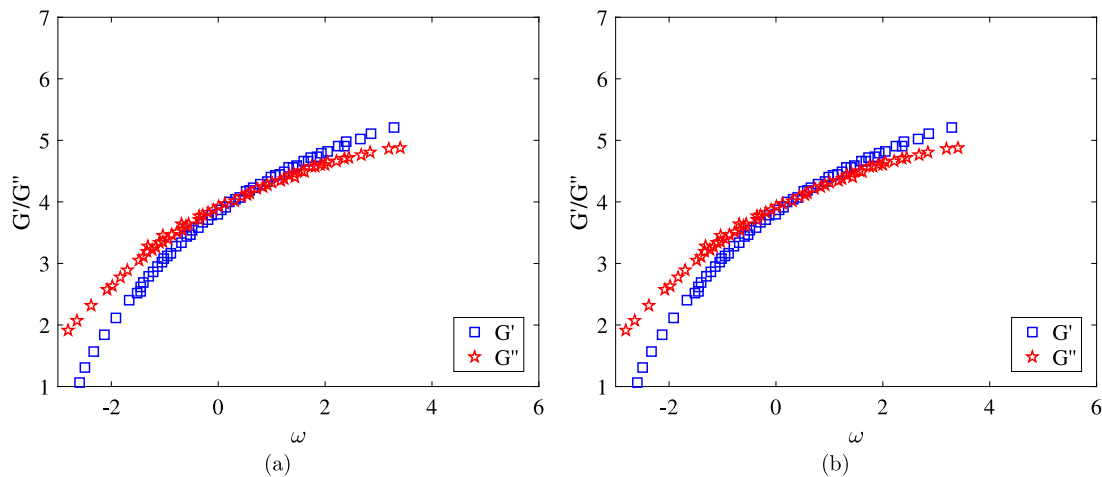


Fig. 4. Experimental data.

were averaged, when applicable. Concerning the parameters controlling MOEA/D algorithm, population size was set to  $k = 100$  individuals (solutions) and the maximum number of generations (iterations) to  $t_{max} = 5000$ , as it was empirically verified that significant improvement is not observed beyond this point. The neighborhood size was set to  $s = 10$  and the maximum number of individuals replaced by offspring to  $\tau_{max} = 1$ . As for the feasible solution space  $\Omega$ , values of  $G_i$  and  $\lambda_i$  were bounded such that  $\log_{10}(G_i) \in [-4, 10]$  and  $\log_{10}(\lambda_i) \in [-6, 6]$  for  $i = 1, \dots, n$ , respectively. A minor customization of Algorithm 1 was implemented which, with probability  $1 - \delta$ , the offspring of  $x^j$  is generated out from the entire population  $P$  instead of from its neighbor solutions ( $B^j$  is substituted with  $1, 2 \dots k$  in line 12, and  $s$  with  $k$  in line 14) – in the reported experiment,  $\delta$  was set to 0.9. All these values were empirically tuned so as to speed up convergence, and should not interfere with the overall numerical result.

In some experiments, we analyzed the results through the hypervolume indicator [30]. As outlined in Section 3, this metric measures the volume of the objective space that is dominated by an approximation set and is bounded by a reference point. It can be defined as the Lebesgue measure  $\Lambda$  of the union of hypercuboids in the objective space as

$$I_H = \Lambda \left( \bigcup_{\mathbf{a} \in A} \{f_1(\mathbf{a}'), \dots, f_m(\mathbf{a}') : \mathbf{a} < \mathbf{a}' < \mathbf{r}\} \right) \quad (33)$$

where  $A = \{\mathbf{a}_1, \dots, \mathbf{a}_{|A|}\}$  is an approximation set and  $\mathbf{r}$  is an appropriately chosen reference point. The hypervolume indicator can measure both the convergence and diversity of  $A$ . Higher values of  $I_H$  are preferable. For the sake of clarity, the normalized values of  $I_H$  will be presented.

With the algorithm configured, MOEA/D implementation was fed with LDPE and PS experimental data and a set of computational experiments were carried out as detailed in the sections that follow.

## 7. Scalarization and variation operators

When applying evolutionary algorithms to solve real-world problems, a few algorithm design decisions must be addressed. Those include the choice of appropriate operators and parameter settings. It is known that there is no general optimization strategy that works the best for all the problems and, therefore, the issue is typically approached through experimentation. In this study, we investigated the impacts of variation and scalarization schemes when applying MOEA/D to the running problem of determination

the relaxation spectrum. We considered three different variants for both schemes introduced in Section 5. The values of the parameters were based on those frequently used in the literature and set as follows.

With respect to variation operators, SBX was used with the crossover probability of  $p_c = 1.0$  and the distribution index of  $\eta_c = 20$ ; DE was used with the crossover probability of  $CR = 1.0$  and the scaling parameter of  $F = 0.5$ ; GM depends only on the size of neighborhood defined in MOEA/D. All these operators are used in conjunction with the PM, with mutation probability of  $p_m = 1/n$  and the distribution index of  $\eta_m = 20$ .

Testing the several combinations of variation and scalarization operators, we have empirically observed that the choice does not drastically impact the result, with an apparent slight superior performance of the combination of SBX and WSFM. For the sake of illustration, Fig. 5 depicts the evolution of the normalized hypervolume<sup>11</sup> during the course of optimization when using different variation schemes.

One possible hypothesis to explain the advantage of SBX variant over the other operators is that the method treats each gene in the chromosome separately, what may be useful for the exploration of multimodal functions with weak or no interactions between the variables. This somewhat corresponds to the characteristics of the problem at hand, as the pairs in Eqs. (4) and (5) can be viewed separable and independent from each other. On the other hand, DE generates an offspring by using the difference between the parent chromosomes. Such a mechanism can exhibit invariant properties with respect to the linear transformation of the search space as the interactions between the variables are taken into account. However, this can come with the cost of slowing the convergence. Lastly, GM builds a probability model using a set of population members and uses a random realization of that model to generate the offspring. It can exhibit similar properties to DE. Nevertheless, the convergence can be further slowed down by randomly sampling from the probability density built.

Regarding scalarization operators, although the choice does not explicitly define the way in which new solutions are produced, it does determine the direction of the search by specifying which solutions are more fit to the given problem. The distinct feature of WSFM, which arguably explains its performance, is that the contour lines which defines the directions of search in

<sup>11</sup> The normalization was obtained by dividing the absolute value of the hypervolume, which depends on the arbitrary choice of the reference point, by the largest value.

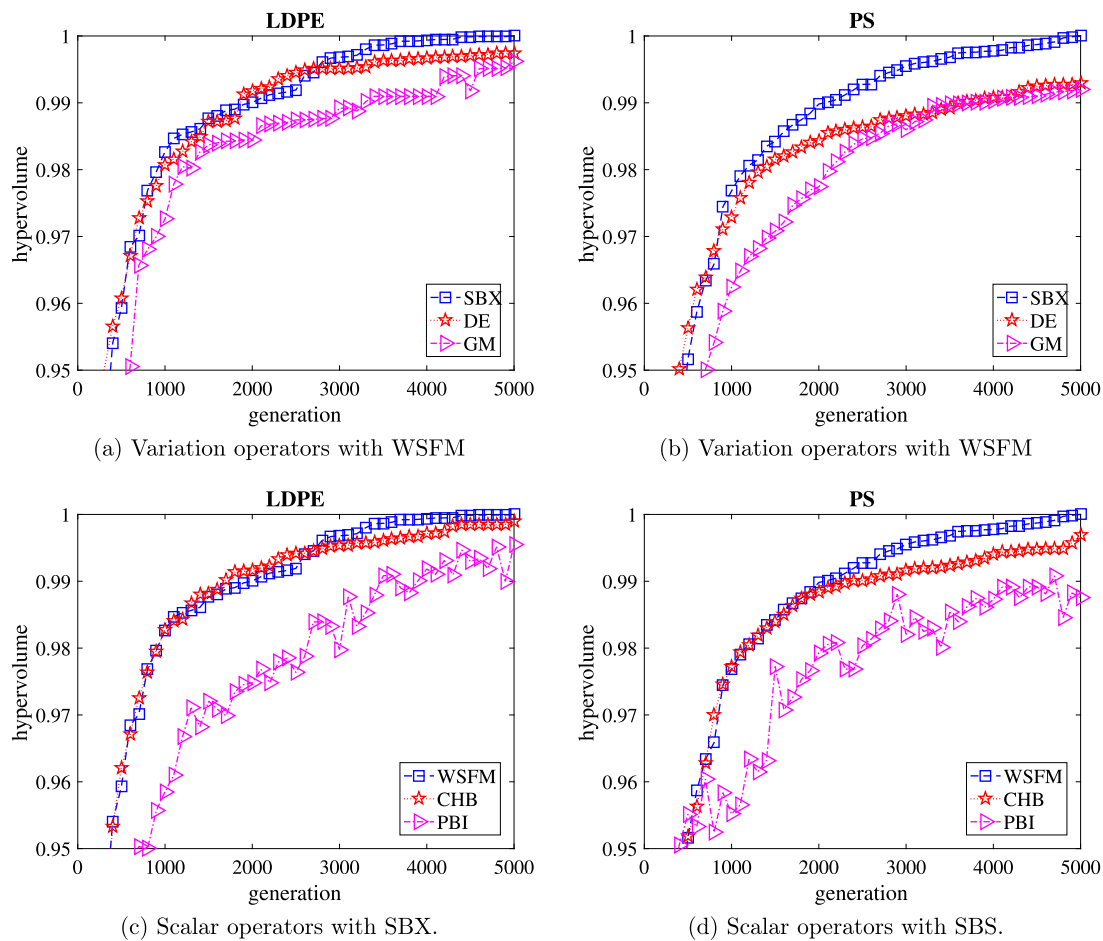


Fig. 5. Evolution of hypervolume for different combination of operators.

Table 1  
Parameter values used in the experiments.

Parameter	Value
Number of trials per experiment	25
Population size $k$	100
Number of generations $t_{max}$	5000
Neighborhood size $s$	10
Maximum replacements $\tau_{max}$	1
Variation operator	SBX
Scalarization operator	WSFM
Number of LDPE data samples	57
Number of PS data samples	109
$\log_{10}(G_i)$	$[-4, 10]$
$\log_{10}(\lambda)$	$[-6, 6]$

MOEA/D are nonlinear [29]. This can be useful not only for appropriately distributing the solutions along the complex geometry of the Pareto front in the final step but also during the whole optimization run. Table 1 lists the parameter values used in the experiments.

For a statistical analysis, we used a nonparametric Wilcoxon rank-sum test at a significance level of  $\alpha = 0.05$ . The results of the test confirmed a statistical difference of the performance of the considered variants of MOEA/D. Similar trends observed for both LDPE and PS can provide support for generalization. Thus, based on the obtained results, we selected the configuration of MOEA/D with SBX and WSFM for the remainder of this paper.

### 7.1. On the number of model parameters

One parameter which does not belong to the method, but to the problem itself, is the number of pairs  $(G_i, \lambda_i)$  chosen to model the physical system under investigation. This accounts for the parameter  $n$  in the solution codification described in Section 1.3. Our first approach to this issue was to empirically evaluate this choice by performing successive runs of the algorithm, starting with  $n = 1$  and incrementing this value at each execution.

For each experiment run (each run consisting in 25 replications with varying random sequences), we computed the average hypervolume of the resulting Pareto set. Fig. 6 shows a graph in which the hypervolume is plotted against  $n$ . As can be noted, the quality of the solution set increases rapidly up to around  $n = 8$  to  $n = 10$ . After this point, no significant improvement is observed by adding more parameters to the relaxation spectrum model. This result is accordant with those reported in the aforementioned paper [17] on the monobjective version of the herein introduced MOEA approach. In the referred article, researchers have similarly found that the regression quality of  $G'$  and  $G''$  curves (using an equally balanced weighted sum scalarization) reached the best quality with no more than 8 to 10 pairs for LDPE, HDPE and polybutadiene experimental data.

The corresponding Pareto sets may be graphically visualized in Fig. 7. The plots show the optimal solutions with reference to regression residuals  $\epsilon_1$  and  $\epsilon_2$  – Eq. (7) and (8) – for the storage and loss moduli, respectively.

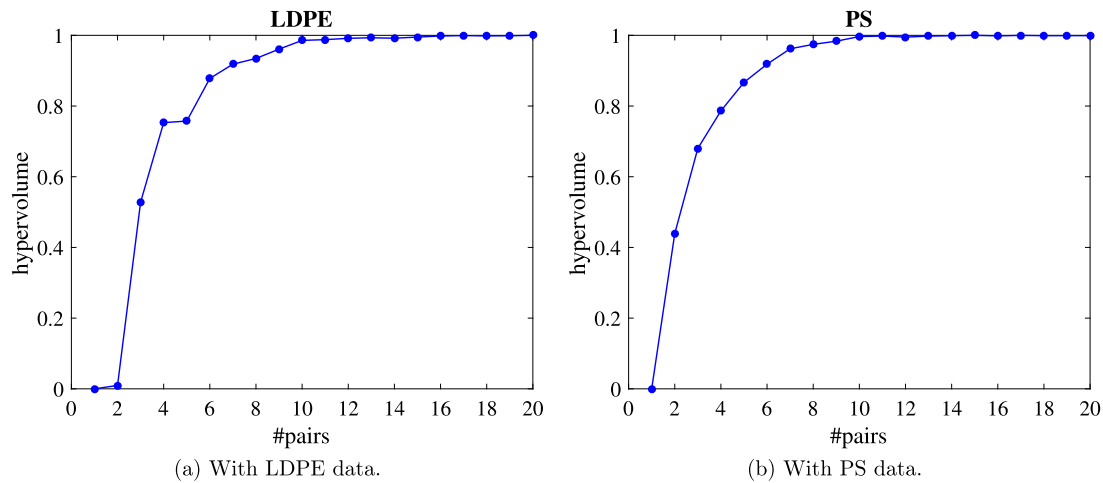


Fig. 6. Normalized hypervolume for varying number of regression parameters.

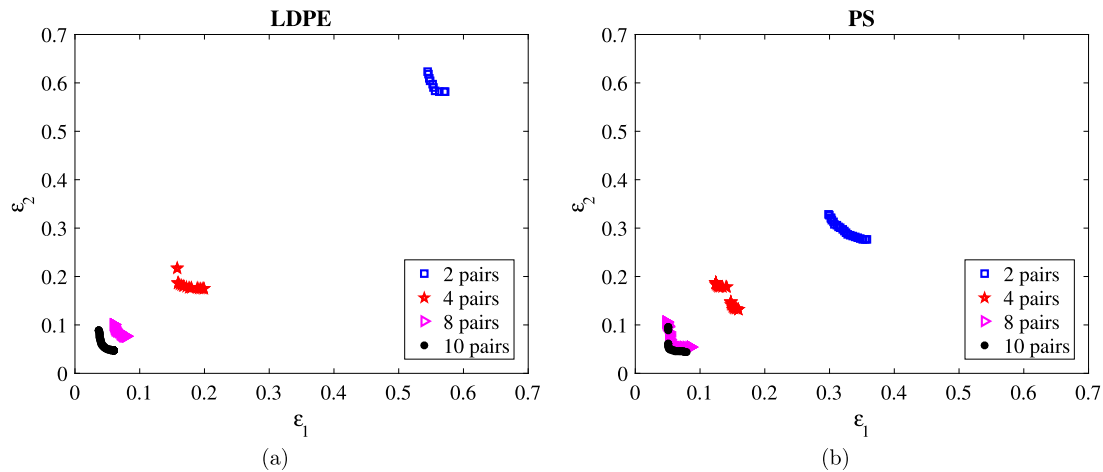


Fig. 7. Pareto fronts for different number of regression parameters.

Corresponding numerical values for the regression residuals may be read in Tables 2 (LDPE) and 3 (PS). The tables list three representative points of the Pareto front, namely, the extremes and the central one:  $S^1$  has the lowest  $\epsilon_1$ ,  $S^1$  has the lowest  $\epsilon_2$  and  $S^m$  has evenly balanced  $\epsilon_1$  and  $\epsilon_2$ .

The  $S^m$  point, referring to the solution in the Pareto front having the smallest distance to (0,0) with respect to  $L_1$  norm, is of particularly interested as it represents intermediate trade-offs between the regression errors. The estimated parameters  $p_i = (G_i, \lambda_i)$  for  $S^m$  are shown on Tables 4 and 5. The list tabulates the values found for  $n = 4, 8, 10$  and  $20$  parameters. The same values may be graphically visualized on the plots of Fig. 8.

### 7.2. Optimal solution length

Up to this point, we have provided evidences that, for the studied materials, no significant gain in regression quality is observable by using more than  $n = 10$  pairs  $(G_i, \lambda_i)$ . This was observed by manually varying the model parameter  $n$  across successive experiment runs. Again, while effective, this expedient does not take full advantage of a multiobjective optimization method such as the one exploited in this study. Indeed, it makes sense to approach the number of parameters as another optimization objective by transforming the problem into a 3-dimensional MOP. In this case, we may have  $\epsilon_1, \epsilon_2$  and  $n$  as our objective functions.

Table 2  
Regression residuals for LDPE with varying number of parameters.

#pairs	$S^1$		$S^m$		$S^2$	
	$\epsilon_1$	$\epsilon_2$	$\epsilon_1$	$\epsilon_2$	$\epsilon_1$	$\epsilon_2$
1	0.5802	0.6227	0.5909	0.5866	0.5957	0.5842
2	0.5446	0.6231	0.5561	0.5837	0.5711	0.5818
3	0.2656	0.4098	0.2956	0.3313	0.3799	0.2991
4	0.1582	0.2174	0.1596	0.1849	0.1991	0.1752
5	0.1491	0.1825	0.1503	0.1804	0.1729	0.1735
6	0.0819	0.1613	0.0931	0.1131	0.1022	0.1097
7	0.0550	0.1148	0.0601	0.0934	0.0697	0.0887
8	0.0725	0.1340	0.0787	0.1068	0.1029	0.1029
9	0.0595	0.1178	0.0669	0.0878	0.0838	0.0833
10	0.0427	0.0828	0.0477	0.0508	0.0603	0.0467
11	0.0414	0.0903	0.0472	0.0564	0.0586	0.0508
12	0.0504	0.0681	0.0573	0.0495	0.0754	0.0448
13	0.0409	0.0726	0.0445	0.0470	0.0549	0.0409
14	0.0446	0.0633	0.0497	0.0457	0.0593	0.0419
15	0.0452	0.0948	0.0505	0.0623	0.0675	0.0577
16	0.0385	0.0684	0.0467	0.0429	0.0525	0.0409
17	0.0444	0.0607	0.0482	0.0459	0.0572	0.0410
18	0.0442	0.0909	0.0491	0.0420	0.0623	0.0382
19	0.0392	0.0541	0.0440	0.0424	0.0526	0.0390
20	0.0436	0.0519	0.0481	0.0420	0.0550	0.0398

To that end, we modified the solution codification described in Section 4 by converting the  $2n$ -length array into a  $3n$ -length one. In this scheme, positions whose indexes are multiples of 3 play

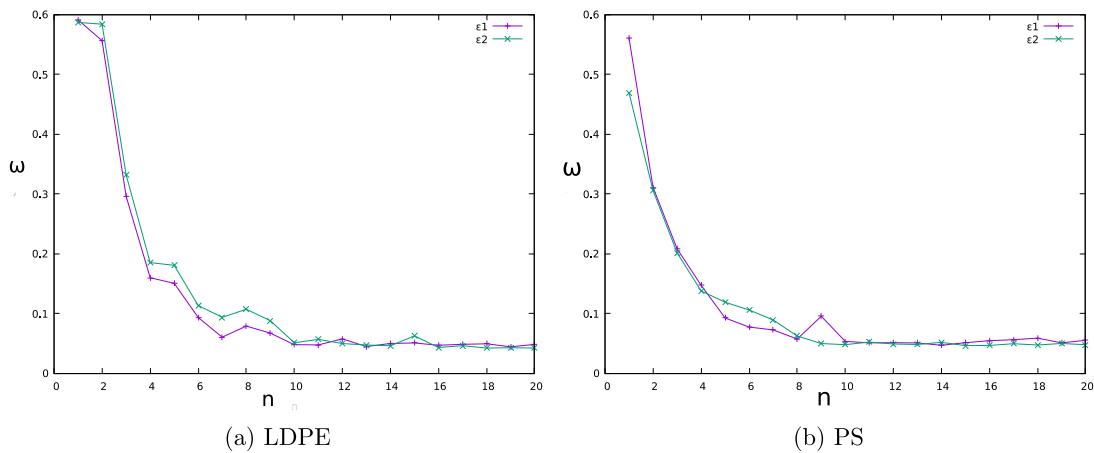


Fig. 8. Regression residuals for vary fixed-length solutions.

Table 3  
Regression residuals for PS with varying number of parameters.

#pairs	$S^1$		$S^m$		$S^2n$	
	$\epsilon_1$	$\epsilon_2$	$\epsilon_1$	$\epsilon_2$	$\epsilon_1$	$\epsilon_2$
1	0.5417	0.5386	0.5602	0.4687	0.5923	0.4576
2	0.2987	0.3285	0.3107	0.3054	0.3573	0.2759
3	0.1953	0.2176	0.2079	0.2012	0.2268	0.1927
4	0.1426	0.1514	0.1470	0.1376	0.1667	0.1322
5	0.0873	0.1338	0.0923	0.1186	0.1591	0.0851
6	0.0705	0.1264	0.0772	0.1056	0.1218	0.0892
7	0.0586	0.1207	0.0723	0.0889	0.1207	0.0717
8	0.0534	0.1077	0.0572	0.0621	0.0779	0.0571
9	0.0927	0.0569	0.0951	0.0495	0.1027	0.0457
10	0.0486	0.0663	0.0531	0.0477	0.0710	0.0433
11	0.0488	0.0865	0.0507	0.0520	0.0630	0.0442
12	0.0467	0.0665	0.0510	0.0488	0.0729	0.0439
13	0.0464	0.0677	0.0505	0.0479	0.0702	0.0436
14	0.0444	0.0614	0.0466	0.0515	0.0642	0.0432
15	0.0469	0.0651	0.0510	0.0461	0.0757	0.0429
16	0.0500	0.0547	0.0543	0.0463	0.0732	0.0438
17	0.0497	0.0819	0.0560	0.0492	0.0802	0.0451
18	0.0516	0.0724	0.0586	0.0471	0.0741	0.0448
19	0.0478	0.0692	0.0507	0.0496	0.0636	0.0437
20	0.0505	0.0671	0.0549	0.0475	0.0686	0.0436

the role of an indicator function  $1_{\mathbb{R}^+}(x_i)$  to determine<sup>12</sup> whether the corresponding pair  $(G_i, \lambda_i)$  preceding it should be used or not in the fitness evaluation—i.e. to fit Eqs. (4) and (5). This is a simple yet effective encoding that allows for the application of the same variation operators we had already been using. No modification in MOEA/D implementation was required.

### 7.3. Loose- $n$ variable-length simulations

As a preliminary exploration of this modified solution encoding, we first rerun the same simulation protocol as before, with only two optimizations of objectives,  $\epsilon_1$  and  $\epsilon_2$ . We left  $n$  a free parameter and did not consider it as an objective in the scalarization function to be minimized. Again, we run the experiment with chromosome lengths varying in the range from 1 to 20. Since some parameters might be disabled, this length is actually an upper bound  $n_{max}$  on the effective number of parameters utilized by the solution.

<sup>12</sup> In mathematics, an indicator function is a function  $1_A : x \rightarrow \{0, 1\} | f(x) = 1$  if  $x \in A$ ,  $f(x) = 0$  otherwise. In this implementation, since encoded values are in  $\log_{10}$  scale, this is equivalent to using the parameter if the respective indicator is positive; and discarding otherwise.

Fig. 9 shows the result, comparing the evolution of the hypervolume along the simulation, between both the fixed-length (preset- $n$ ) and the variable-length (free- $n$ ) solution versions.

In the plot, the abscissa axis indicates the maximum  $n$  allowed in the experiment with the variable-length version, and the exact  $n$  value with the fixed-length version. While the curve for PS material does not add much with respect to the expected results, the plot for the LDPE material may come up somewhat surprising at a first glance. As it can be noticed, up to  $n = 8$ , the Pareto-optimal solutions of the free- $n$  experiment turned out superior than those found by the preset- $n$  variant, for the same allowed number of parameters, consistently across the 25 replications.

Albeit not an immediately intuitive outcome, one conjecture which might be evoked to possibly explain it is that, while the fixed-length simulation was forced to optimize exactly  $n$  model parameters – pairs  $(G_i, \lambda_i)$  – the variable-length variant was allowed to get rid of excess parameters (by disabling them through the indicator gene) if those did not effectively contributed to the regression quality. A similar hypothesis had been already brought about in the paper on the monobjective EA [17], when authors noticed their genetic algorithm implementation's striving to reduce the numeric contribution of excess terms of Eq. (2) when the program was constrained to deal with an unnecessarily large  $n$ .

In order to investigate this finding further, we counted the effective number of model parameters  $n$  in the solution sets found. What we observed (Table 6) is that when the solution was bounded by a length of up to  $n_{max} = 8$ , all parameters were used. On the other hand, when we set  $n_{max}$  to larger values, the obtained Pareto fronts, in different executions, were composed by solutions with varying lengths, ranging from  $n = 10$  to  $n = 14$ .

The interpretation of this result evokes the observation that if a gene does not effectively adds to the individual's competitiveness within the population, there is no bias acting to preserve it along generations. The indicator gene provides a quicker mechanism to “turn off” useless genes, whereas the fixed-length version requires the algorithm to gradually diminish its numerical significance across a longer succession of iterations. Nevertheless, since there is no active selective pressure to shorten the solution length, this effect is resulting solely from random chances and there is a non-negligible possibility that the population converges to some stable configuration, and remains drifting around it indefinitely.

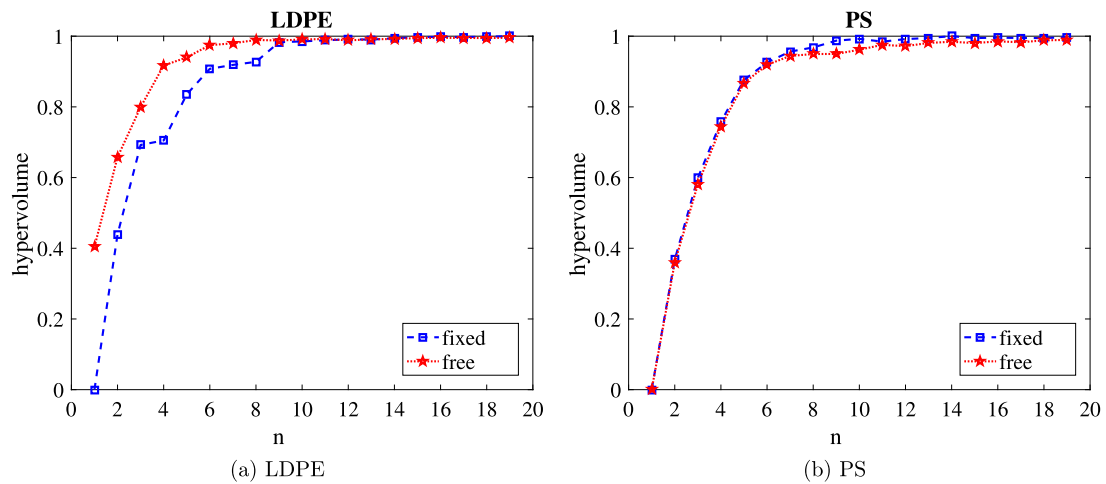
This phenomenon may be illustrated by Fig. 10. The plot shows the  $(G_i, \lambda_i)$  resulting from one of the simulation experiments in this series. For the LDPE material, all Pareto-optimal solutions ended up by having  $n = 12$  parameters, while for PS this

**Table 4**  
Model parameters obtained for LDPE (log<sub>10</sub>) scale.

$p_i$	$n = 4$		$n = 8$		$n = 10$		$n = 20$	
	$G_i$	$\lambda_i$	$G_i$	$\lambda_i$	$G_i$	$\lambda_i$	$G_i$	$\lambda_i$
1	1.0000	3.5027	0.9604	1.7604	0.8821	0.5925	-5.5821	0.9153
2	3.4609	1.0000	0.9999	2.3108	0.9992	0.6121	-0.9407	2.2185
3	4.3533	-0.4030	1.0000	1.7972	1.0000	2.3973	0.6668	1.9350
4	4.9083	-1.9238	2.3007	1.9356	2.0982	2.0332	0.7597	0.5326
5	-	-	3.5052	0.8408	2.9220	1.2129	0.8954	1.9620
6	-	-	4.2288	-0.3131	3.4530	0.6338	0.9945	1.9425
7	-	-	4.7190	-1.6083	4.0476	-0.2074	0.9954	-1.3085
8	-	-	5.1976	-2.9576	4.2465	-0.9987	0.9959	2.0002
9	-	-	-	-	4.7048	-1.8323	1.9193	2.1605
10	-	-	-	-	5.1750	-3.0187	2.8544	0.4624
11	-	-	-	-	-	-	3.0138	1.2050
12	-	-	-	-	-	-	3.1261	0.4908
13	-	-	-	-	-	-	3.2761	0.4748
14	-	-	-	-	-	-	3.5745	-0.8621
15	-	-	-	-	-	-	3.6571	-0.9397
16	-	-	-	-	-	-	4.0707	-0.3342
17	-	-	-	-	-	-	4.4849	-1.5109
18	-	-	-	-	-	-	4.5995	-2.1680
19	-	-	-	-	-	-	4.9211	-2.8951
20	-	-	-	-	-	-	7.2341	-5.9490

**Table 5**  
Model parameters obtained for PS (log<sub>10</sub>) scale.

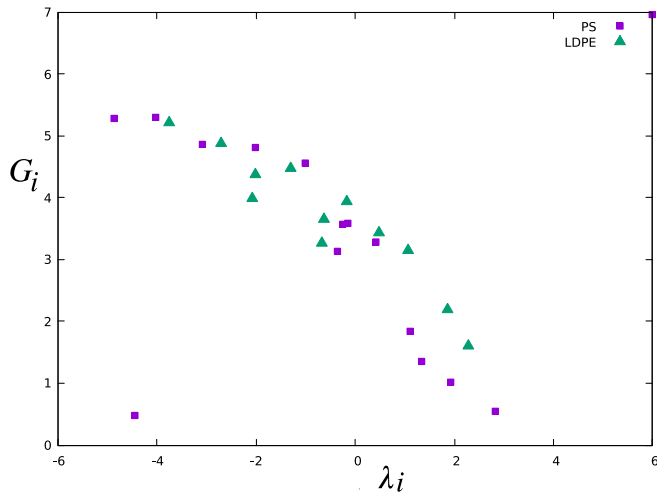
$p_i$	$n = 4$		$n = 8$		$n = 10$		$n = 20$	
	$G_i$	$\lambda_i$	$G_i$	$\lambda_i$	$G_i$	$\lambda_i$	$G_i$	$\lambda_i$
1	3.2705	0.5701	0.9733	3.2983	1.1559	2.2270	-3.9444	-2.3928
2	4.3099	-0.4668	1.7121	1.3521	2.4308	0.9672	-3.3438	-5.7345
3	4.8249	-1.5486	3.2197	0.4859	3.5956	-0.2687	-0.6443	-2.3102
4	5.2668	-3.1727	4.0032	-0.2402	3.6534	0.1424	0.6861	4.7101
5	-	-	4.6213	-1.0809	3.9563	-2.6649	0.9157	2.4839
6	-	-	4.8890	-2.2724	4.5326	-0.9294	2.0903	1.0194
7	-	-	5.2745	-3.6566	4.7608	-1.9399	2.0907	0.8583
8	-	-	6.5001	-5.4284	4.8948	-3.0658	2.1622	0.8049
9	-	-	-	-	5.3968	-4.1633	3.3664	-0.5471
10	-	-	-	-	6.7980	-5.7959	3.5641	-0.5705
11	-	-	-	-	-	-	3.6659	-0.5189
12	-	-	-	-	-	-	3.6998	-1.7882
13	-	-	-	-	-	-	3.7480	-4.3482
14	-	-	-	-	-	-	3.7625	0.0597
15	-	-	-	-	-	-	4.5835	-1.1875
16	-	-	-	-	-	-	4.7864	-2.2198
17	-	-	-	-	-	-	5.0387	-3.3672
18	-	-	-	-	-	-	5.1223	-5.5186
19	-	-	-	-	-	-	5.3655	-4.3210
20	-	-	-	-	-	-	6.9716	-6.0000



**Fig. 9.** Hypervolume evolution for bounded free- $n$  and preset- $n$ .

**Table 6**  
Effective  $n$  for loose- $n$  variable-length simulations.

Allowed $n$	Effective $n$
4	4
6	6
8	8
10	9
20	14



**Fig. 10.** Model parameters for a loose- $n$  variable-length simulation.

number was  $n = 15$  (regression residuals were minimally higher but compatible to the ones listed in Tables 2 and 3). With a very small mutation probability, this result has little chance of changing solely by random fluctuations. Are these the shorter solutions with the given regression quality? Based on the plot, chances are that the answer can be negative. Indeed, bearing in mind that each parameter (dot on the plot) corresponds to a summand in Eq. (2) it is easy to recognize that the two outliers at the extremes of the scales (bottom-left and top-right corners) have little numerical impact, since either a small  $G_i$  or a large  $\lambda_i$  minimizes the term. Furthermore, if  $\lambda_i \approx \lambda_j$ , then the two corresponding addends may be approximated by  $(G_i + G_j)e^{(t-t')/\lambda_i}$ . Upon this reasoning, it is suggestive that there may be room for improvement regarding the optimization of the number of parameters. Adding a selective pressure to minimize  $n$  is the goal of the experiment which follows.

#### 7.4. Three-objective formulation

Having tested the modified solution encoding with the indicator gene, we then designed a three-dimensional MOP, with the objectives of minimizing  $\epsilon_1$ ,  $\epsilon_2$  and  $n$  simultaneously. With the same algorithm setup and input data we rerun the experiment for LDPE, obtaining the result depicted in Fig. 11.

At the top (Fig. 11(a)), the plot shows the tridimensional Pareto surface with the distribution of the optimum solutions in the objective-function space; for the sake of clarity, only a single replication of the experiment was plotted. In this simulation, the best solutions regarding regression quality (lower  $\epsilon_1$  and  $\epsilon_2$ ) use  $n = 6$  parameters; they appear closely agglomerated at the top and are difficult to discern. Figs. 11(b) and 11(c) show the projection of the Pareto surface on the  $\epsilon_1 \times n$  and  $\epsilon_2 \times n$  planes, respectively.

What this result is showing is that lower values of  $n$  produce lengthier Pareto fronts, i.e. a wider range of choices of regression parameters with varying relative combinations of  $\epsilon_i$  and  $\epsilon_2$ , either

**Table 7**  
Comparison with other methods.

Method	# pairs	$\epsilon_1$	$\epsilon_2$
Laun [6]	8	0.0556	0.0591
GenFit [17]	8	0.0431	0.0410
GenFit [17]	10	0.0431	0.0410
Our method	10	0.0429	0.441

larger than what is possible with larger solution lengths (large  $n$ ). Pareto fronts with lower regression errors, on the other hand, are shorter and better balanced with respect to regression errors in both storage and loss moduli. The resulting Pareto sets on the regression residuals plane may be better visually appreciated in Fig. 12.

In this experiment, we maintained the same algorithm setup than before, with a population of size  $k = 100$  to be evolved along  $\tau_{max} = 5000$  generations. It should be noted, therefore, that now, with three objectives, Pareto-optimal solutions which remain in the final set due to a dominance on parameter length (lower  $n$ ) compete with the other solutions which would be preserved based on superior regression error (lower  $\epsilon_1$  or  $\epsilon_2$ ). Therefore, the “chance of survival” of some good solutions with respect to regression quality is “stolen” from the finite population by those with shorter lengths. Likewise, the survival of solutions with larger  $n$  may be compromised by dominance on  $\epsilon_1$  or  $\epsilon_2$ . The consequence is that, the same population size and number of generations have their potential to explore the solution search-space diminished.

A straightforward counter measure for this scenario is to increase those values. We then rerun the experiment to allow  $k = 300$  solutions in the population and optimize them along  $\tau_{max} = 15000$  generations. With more freedom to explore the solution space, we obtained the best solutions with balanced regression quality with  $n = 10$  parameters, and regression residuals of  $\epsilon_1 = 0.480$  and  $\epsilon_2 = 0.492$ , very close to what was obtained with the fixed-length experiment with  $n = 10$  parameters—shown in Table 4.

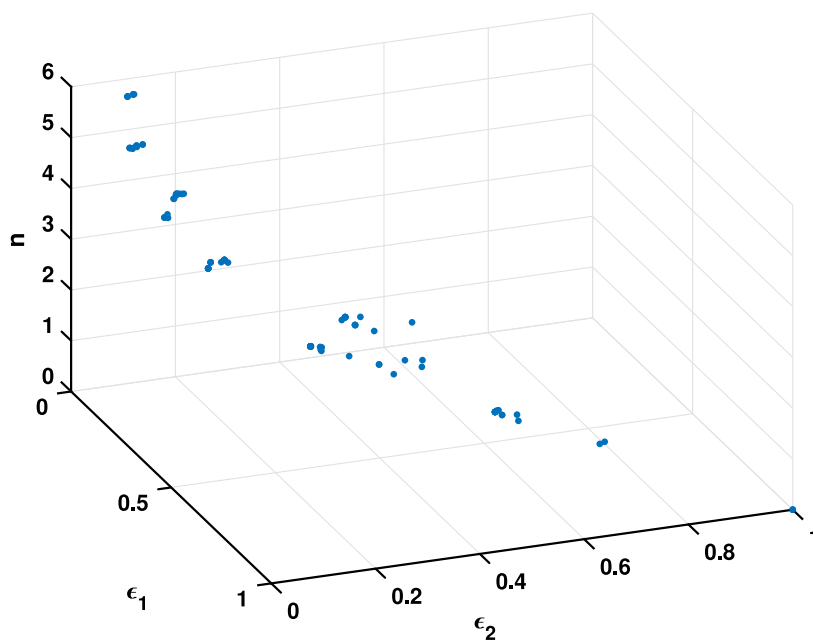
### 8. Assessment of the regression quality

Founded on the rationales and empirical evidences discussed in previous sections, a reasonably adequate configuration to assess the quality of the Pareto-optimal solutions obtained by our implemented method is one which minimizes  $\epsilon_1$  and  $\epsilon_2$ , with a parsimonious fixed-length solution encoding. This not only speeds up the convergence but reduces the burden of optimizing excess parameters, requiring thus both a smaller size population and smaller number iterations.

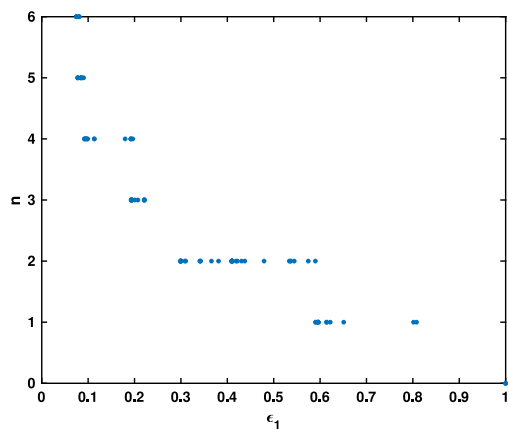
Figs. 13(a) and 13(b) depict the result obtained for the LDPE material. The model was fit with  $n = 10$  (empirical lower bound of  $\epsilon$  suggested by Fig. 8) parameters from Table 4. Plots compare the experimental data and the corresponding values forecast by the fitted model.

The regression residuals for LDPE material are, respectively  $\epsilon_1 = 0.477$  and  $\epsilon_2 = 0.0508$ , comparable to the best results reported in the literature so far. In a subsequent run of the same experiment, for a much larger number of iterations ( $t_{max} = 50,000$ ), we could obtain even lower regression residuals for  $n = 10$ , reaching  $\epsilon_1 = 0.0429$  and  $\epsilon_2 = 0.0441$ —practically equivalent to those reported by authors [17] of the monobjective version of the EA method. This result is summarized in Table 7.

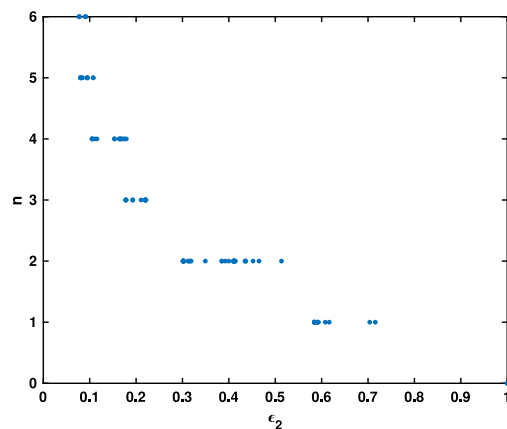
As mentioned, the said monobjective version by the author already outperformed several analytical techniques to which it was compared in the cited work [17], what indicates that our multiobjective evolutionary approach is deserving the appreciation as a promising tool in the modeling of polymer viscosity



(a) The 3D objective. space.

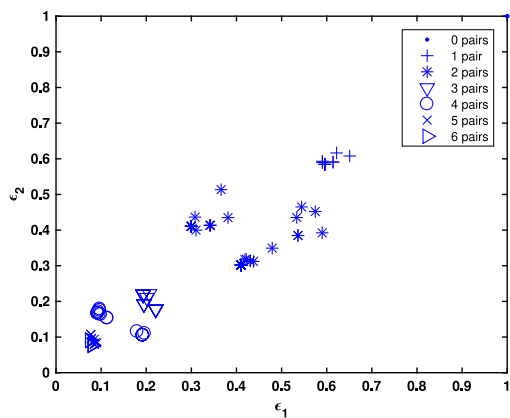


(b) On  $e_1 \times n$  plane.

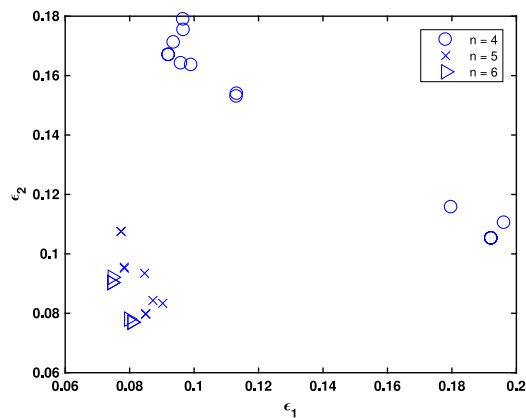


(c) On  $e_2 \times n$  plane.

Fig. 11. Solutions of a free- $n$  simulation in the objective space.



(a) Solutions of all lengths.



(b) Solutions with  $n = 4, 5$  and  $6$

Fig. 12. Regression residuals of varying bounded free- $n$  simulations.

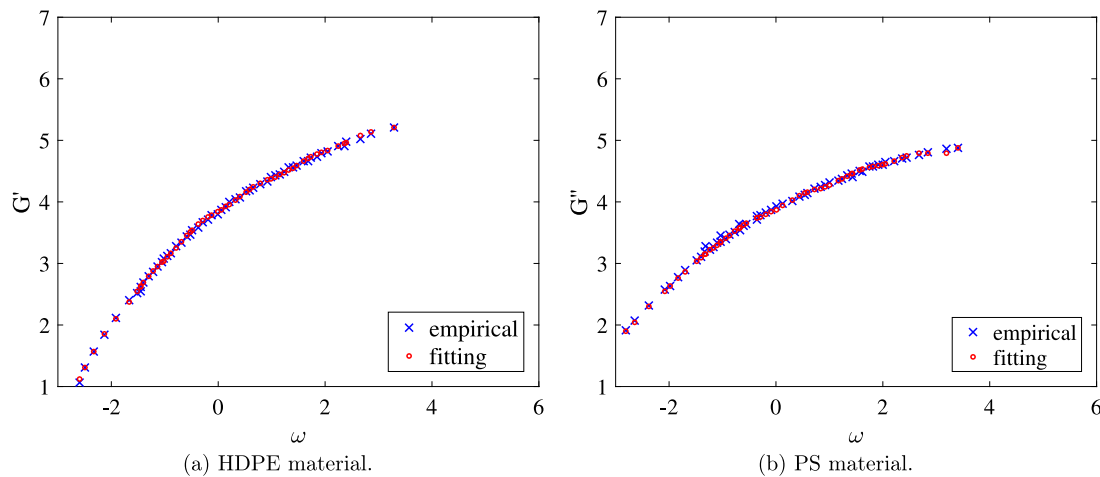


Fig. 13. Fitting curves for a bounded free- $n$  simulation with  $n = 8$ .

models. It is also relevant to bear in mind that, while the herein introduced multiobjective method achieves similar superiority, its key proposed contribution is not necessarily to reduce the regression residuals even further, but rather to extend the solution from a mono to a multiobjective perspective. In the monoobjective approach, the relative weights of either residuals ( $\eta_1, \eta_2$ ) are set *a priori* and one single solution which minimizes their linear combination is found. If, later on, another set of weights is to be considered, another full execution of the optimization algorithm is needed. In the multiobjective version, on the other hand, a whole set of solutions with varying tuples  $(w_1, w_2)$  is evolved at once, each corresponding to a different set of weights. The choice of which solution is more interesting is an *a posteriori* decision, to be addressed by the decision maker based on what is more important to prioritize in each case.

One noteworthy remark about this result is that, in any regression problem, unrealistic oscillatory characteristic of forecast model due to overfitting is inherently associated to excess parameters. While some deterministic methods aim at improve smoothness using arbitrary assumptions such as regularization or constraints on parameters, our evolutionary adaptive method provides the basis for automatically reducing excesses parameters to an adequate bound, relying only on data's intrinsic information, thereby naturally optimizing curve smoothness.

## 9. Conclusions

The process of deriving a causal model to describe a phenomenon of interest mathematically translates into an inverse problem. In both scientific and industrial research, the associated regression analysis frequently involves conflicting goals of minimizing the residuals on each individual dependent variable, what is often approached as a monoobjective optimization problem in which one wishes to minimize an aggregate of those errors as quantified by a scalarization function, and where the weights are chosen by *a priori* importance assignment. More often than not, the order of the model, given by the number of regression parameters, is not well known in advance from the theory and must be set empirically. In addition, deterministic numerical methods underlying those techniques often rely on assumptions extrinsic to the problem such as regularization constants and restrictions on parameter distribution. All those non-inherent factors represent sources of model biases. This paper introduces a multiobjective optimization (MOP) method for nonlinear regression analysis, which is capable of simultaneously minimizing the model order and estimating parameter values without need of exogenous regularization constraints. The proposed non-deterministic strategy,

based on evolutionary computational methods, aims at finding the Pareto-front of all optimal solutions with respect not only to individual regression errors, but also to the number of parameters needed to fit the data, automatically reducing the model order. The approach does not require arbitrary constraints on objective weights, regularization parameters, or other exogenous assumptions to handle the ill-posed inverse problem. Whilst the main ideas are introduced through a case study, the method is general and applicable to other problem domains, as it is discussed in the sections explaining the rationales, implementation, simulation experiments, and comparison with other methods, with experimental evidences that it can outperform state-of-art techniques. For the interest of a multidisciplinary audience, the article lays a detailed introduction of essential concepts and techniques.

The determination of the discrete linear relaxation spectrum of polymers is a relevant subject both in scientific and industrial extents, where small-amplitude oscillatory shear is one important experimental method for this purpose. The experiment produces a series of measurements quantifying the material's storage and relaxation moduli at varying frequencies. Those experimental data then are used to determine a set of regression parameters that should simultaneously fit the two equations describing the material properties. What makes this problem particularly challenging is that it brings about an ill-posed, nonlinear problem, with two conflicting objectives. Most state-of-art research results and also state-of-practice methods suffer from several limitations resulting from the constraints imposed by analytical methods. Some require the artificial addition of excess degrees of freedom, others include arbitrary parameters, others still impose constraints that rule out viable solutions. In many cases, the solution is overly complex or approximated by heuristics, and results exhibit unrealistic features not consistent with the real material.

As an alternative, [17] proposed a non-deterministic optimization method based on evolutionary algorithms (EA) which does not rely on extrinsic knowledge or arbitrary constraints. EAs can handle ill-posed problems naturally, as they do not build up neither on linearity, nor on continuity hypotheses concerning the solution or objective spaces. EAs are based on an educated trial-and-error iterative strategy, through which a collection of prospective solutions are optimized in parallel by means of recombination and probabilistic best-fit selection. This characteristic also allows EAs to overcome premature convergence to local optima and to speed up the search of the global optimum. In their experimental study Monaco et al. [17], the authors successfully applied a genetic algorithm to determine the relaxation spectrum



of a few polymers. The EA implementation was capable of outperforming the regression quality obtained by other methods, and with a lower number of model parameters. The goal of minimizing the regression residuals of both the storage and loss moduli are nevertheless conflicting objectives and, not unlike most of the related works described in the literature, the referred EA implementation approached the issue as a monobjective problem—by tackling it as the minimization of an aggregate function of the residuals, pondered by an arbitrary set of weights.

In this paper, we go a step further, exploiting the generalizable EA metaheuristics to face the same challenge as a strictly multiobjective problem. In other words, we introduce a method to obtain several Pareto-optimal solutions, leaving the choice of objective weights for the decision making analyst. We designed our method as a multiobjective evolutionary algorithm based on decomposition, due its convenient capabilities of generating a fair distribution of representative Pareto-optimal solutions and the possibility of using existing, well-known monobjective variation operators. The rationales and implementation aspects are explained in detail and a set of experiments are reported, which demonstrates the capabilities and limitations of the proposed method.

The comparison of the results of the proposed method with those of related works is not straightforward, seeing that our method produces a set of (rather than a single) solutions of varying qualities regarding the two analyzed objectives—it is not immediate which solution of the set should be taken as a basis for comparison. For the sake of practical illustration, though, the solutions with nearly equal regression residuals for both storage and loss moduli we obtained are numerically close to that reported in the referenced literature. In particular, we were able to approximately reproduce the results obtained by the mentioned monobjective EA [17] for the LDPE material with respect to both regression errors and number of model parameters (indeed, the literature [31] suggests that multiobjective optimization methods can outperform monobjective analogs in virtue of the lower scale sensitivity and susceptibility to local-optima traps—although subject to the limitations of the intrinsic experimental errors in the data sample). Like in the said experiment, the solutions produced by MOAD/D have similar or lower regression residuals than those yielded by deterministic methods reported in the literature (Section 1.2), and with a lower number of parameters.

The exact sets of  $(G_i, \lambda_i)$  to fit the materials used as case studies are not the central contribution aimed by this paper but, rather, the introduction of a method to generate the Pareto-optimal solutions. As far as our knowledge goes, there is no other proposed method in the literature which approaches the determination of the linear discrete relaxation spectrum of polymers as a multiobjective problem, and allows for addressing the compromise between the regression qualities of both storage and loss moduli as *a posteriori* decision making, up to the specialist's discretion and relevant criteria for the specific application. Nonetheless, for the purpose of comparison with other works, we base our experiments on data sets referred in the literature [17]. Should more accurate data sets produced by more modern equipment be analyzed, this can be easily achieved by applying the outlined algorithm implementation to produce reference parameter tables for practical use. Moreover, it is worth to note that unlike deterministic methods, which produce a unique result for a given data set, evolutionary algorithms are inherently non-deterministic and, therefore, shall produce different results at each run, even for the same data set—as a result of exploring the solution search space through different paths. The convergence property, however, implies that solutions tend to result equivalent in terms of the optimization objectives. Actually, while we have limited our experiments by a maximum number

of iterations, a bound on the maximum regression error or else on the incremental optimization gain at each iteration may be alternatively used as stop criteria, instead.

One noteworthy outcome of our experiment is that under less constrained conditions, and when the algorithm is left free to choose the solution's length, it naturally tends to reduce excess parameters, contributing to reduce regression residuals and improve curve smoothness simultaneously, based solely on the data's intrinsic information and without the need to *a priori* assumption. As already pointed out, it is thus suggestive to ponder on whether the number of parameter models needed to represent the material's behavior may have some inherent physical meaning. In our experiments the solutions's length converged to around 8 to 10 parameters, what is consistent to previous findings by [17]. In our study we considered the regression residuals as the primary quality metric to assess the results. Yet, we opportune remarked that this is not the only parameter to evaluate a solution for this problem, as overfitting may produce low-residual solutions with quite unrealistic relaxation spectra such as with pathological oscillatory characteristics. One important feature of the proposed method to be highlighted is that the solution quality measure is not intrinsic to it as it is the case of several analytical techniques. In the EA approach, differently, the role of evaluating solutions' quality is played by the fitness function, which is a generic black-box for implementing any quality criteria—wherefrom EA being considered a metaheuristic. Put shortly, this comes from the algorithm core being the repeated iteration of a sequence of steps: (a) evaluate all the solutions found so far in terms of how they fit the given criteria; (b) select (proabilistically) some best solutions; (c) produce some new solutions by performing (small) random changes on the selected best-fitted; (d) if some of newly created solutions are better than those in the original pool, replace the latter with the former. The procedure starts with a randomly generated solution pool and ends upon reaching some stop criteria. In step (a), the criterion to evaluate a solution is any function. The method implementation may leave it as a placeholder which the user may later fill in with any function they desires. This does not modify neither the algorithm nor the rest of the implementation. We have used Eq. (6) for fitness assignment; the domain analyst is free to choose other, possibly more sophisticated, criteria. The set of objectives of this MOP method are therefore user-defined, rather than inherent to the method, what yields great flexibility.

We believe both the discussed rationales and provided experimental evidences demonstrate the proposal of evolutionary multiobjective optimization as a powerful resource for nonlinear regression analysis applied to the modeling of causal phenomena, which is useful not only to technique-unbiased estimation of model parameters, but also to optimize the model order and investigate its structure, what may bring about implications for the theoretical understanding of its nature.

### CRedit authorship contribution statement

**Francisco José Monaco:** Methodology, Validation, Formal analysis, Investigation, Visualization, Writing – review & editing. **Roman Denysiuk:** Software, Investigation, Data Curation. **Alexandre Claudio Botazzo Delbem:** Conceptualization, Methodology, Supervision. **António Gaspar-Cunha:** Conceptualization, Methodology, Resources, Project administration, Funding acquisition.

### Declaration of competing interest

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

## Acknowledgments

This work is funded by National Funds through FCT - Portuguese Foundation for Science and Technology, References UIDB/05256/2020 and UIDP/05256/2020 and the European project MSCA-RISE-2015, NEWEX, Reference 734205.

## References

- [1] N.W. Tschoegl, *The phenomenological theory of linear viscoelastic behavior*, Springer-Verlag, Berlin, 1989.
- [2] J.M. Dealy, K.F. Wissbrun, *Melt Rheology and its Role in Plastics Processing*, Springer US, Boston, MA, 1990, <http://dx.doi.org/10.1007/978-1-4615-9738-4>, URL: <http://link.springer.com/10.1007/978-1-4615-9738-4>.
- [3] N. Orbey, J.M. Dealy, Determination of the relaxation spectrum from oscillatory shear data, *J. Rheol.* 35 (6) (1991) 1035–1049.
- [4] C. Friedrich, B. Hofmann, Nichtkorrekte aufgaben in der rheometrie, *Rheol. Acta* 22 (5) (1983) 425–434, <http://dx.doi.org/10.1007/BF01462929>, URL: <http://link.springer.com/10.1007/BF01462929>.
- [5] J. Honerkamp, J. Weese, Determination of the relaxation spectrum by a regularization method, *Macromolecules* 22, No. 11 (1989) 4372–4377.
- [6] H.M. Laun, Description of the non-linear shear behaviour of a low density polyethylene melt by means of an experimentally determined strain dependent memory function, *Rheol. Acta* 17 (11) (1978) 1–15.
- [7] M. Baumgaertel, H. Winter, Determination of discrete relaxation and dynamic mechanical data, *Rheol. Acta* 28 (1989) 511–519.
- [8] D.H.S. Ramkumar, J.M. Caruthers, H. Mavridis, R. Shroff, Computation of the linear viscoelastic relaxation spectrum from experimental data, *J. Appl. Polym. Sci.* 64 (1997) 2177–2189.
- [9] T. Roths, D. Maier, C. Friedrich, M. Marth, J. Honerkamp, Determination of the relaxation time spectrum from dynamic moduli using an edge preserving regularization method, *Rheol. Acta* 39 (2) (2000) 163–173, <http://dx.doi.org/10.1007/s003970050016>, URL: <http://link.springer.com/10.1007/s003970050016>.
- [10] S. Gerlach, A. Matzenmiller, Comparison of numerical methods for identification of viscoelastic line spectra from static test data, *Internat. J. Numer. Methods Engrg.* 63 (2005) 428–454.
- [11] I. Emri, N.W. Tschoegl, Determination of mechanical spectra from experimental responses, *Int. J. Solids Struct.* 32 (6–7) (1995) 817–826, [http://dx.doi.org/10.1016/0020-7683\(94\)00162-P](http://dx.doi.org/10.1016/0020-7683(94)00162-P), URL: <https://www.sciencedirect.com/science/article/pii/002076839400162P>.
- [12] Zatloukal, J. Vlcek, C. Tzoganakis, T. Dobbie, Computation of the linear viscoelastic relaxation spectrum from capillary viscosity data, *Adv. Polym. Technol.* 19 (4) (2000) 277–286.
- [13] J.D. Guzmán, J.D. Schieber, R. Pollard, A regularization-free method for the calculation of molecular weight distributions from dynamic moduli data, *Rheol. Acta* 44 (4) (2005) 342–351, <http://dx.doi.org/10.1007/s00397-004-0414-3>, URL: <http://link.springer.com/10.1007/s00397-004-0414-3>.
- [14] A.Y. Malkin, The use of a continuous relaxation spectrum for describing the viscoelastic properties of polymers, *Polym. Sci. Ser. A* 48 (1) (2006) 39–45, <http://dx.doi.org/10.1134/S0965545X06010068>, URL: <http://link.springer.com/10.1134/S0965545X06010068>.
- [15] S.H. Lee, J.-E. Bae, K.S. Cho, Determination of continuous relaxation spectrum based on the Fuoss-Kirkwood relation and logarithmic orthogonal power-series approximation, *Korea-Aust. Rheol. J.* 29 (2) (2017) 115–127, <http://dx.doi.org/10.1007/s13367-017-0013-3>, URL: <http://link.springer.com/10.1007/s13367-017-0013-3>.
- [16] R. Anderssen, A. Davies, F. de Hoog, R. Loy, Derivative based algorithms for continuous relaxation spectrum recovery, *J. Non-Newton. Fluid Mech.* 222 (2015) 132–140, <http://dx.doi.org/10.1016/J.JNNFM.2014.10.004>, URL: <https://www.sciencedirect.com/science/article/pii/S037702571400192X?via%3Dihub>.
- [17] F.J. Monaco, A.C.B. Delbem, F.M. Federson, Genetic algorithm for the determination of linear viscoelastic relaxation spectrum from experimental data, *J. Appl. Polym. Sci.* 113 (1) (2009) 122–135, <http://dx.doi.org/10.1002/app.29496>, URL: <http://doi.wiley.com/10.1002/app.29496>.
- [18] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Massachusetts, 1989.
- [19] H. Tamaki, H. Kita, S. Kobayashi, Multi-objective optimization by genetic algorithms: a review, in: *Proceedings of IEEE International Conference on Evolutionary Computation*, IEEE, 2002, pp. 517–522, <http://dx.doi.org/10.1109/icec.1996.542653>, URL: <http://ieeexplore.ieee.org/document/542653/>.
- [20] L. Miguel Antonio, C.A. Coello Coello, Coevolutionary multiobjective evolutionary algorithms: Survey of the state-of-the-art, *IEEE Trans. Evol. Comput.* 22 (6) (2018) 851–865, <http://dx.doi.org/10.1109/TEVC.2017.2767023>, URL: <https://www-sciencedirect.com/science/article/pii/S2210650211000058>.
- [21] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach, *IEEE Trans. Evol. Comput.* 3 (4) (1999) 257–271, <http://dx.doi.org/10.1109/4235.797969>, URL: <http://ieeexplore.ieee.org/document/797969/>.
- [22] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, *IEEE Trans. Evol. Comput.* 11 (6) (2007) 712–731, <http://dx.doi.org/10.1109/TEVC.2007.892759>, URL: <http://ieeexplore.ieee.org/document/4358754/>, arXiv:1701.00879.
- [23] K. Deb, R.B. Agrawal, Simulated binary crossover for continuous search space, *Complex Syst.* 9 (1995) 115–148.
- [24] H. Li, Q. Zhang, Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II, *IEEE Trans. Evol. Comput.* 13 (2) (2009) 284–302.
- [25] A. Zhou, Q. Zhang, G. Zhang, A multiobjective evolutionary algorithm based on decomposition and probability model, in: *Proc. Congr. Evol. Comput.*, 2012, pp. 1–8.
- [26] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*, in: *Wiley-Interscience Series in Systems and Optimization*, John Wiley & Sons, 2001.
- [27] K. Miettinen, *Nonlinear Multiobjective Optimization*, in: *International Series in Operations Research and Management Science*, vol. 12, Kluwer Academic Publishers, 1999.
- [28] Q. Zhang, H. Li, MOEA/D: a multiobjective evolutionary algorithm based on decomposition, *IEEE Trans. Evol. Comput.* 11 (6) (2007) 712–731.
- [29] R. Denysiuk, A. Gaspar-Cunha, Weighted stress function method for multiobjective evolutionary algorithm based on decomposition, in: H. Trautmann, et al. (Eds.), *LNCS*, Vol. 10173, Springer, Cham, 2017, pp. 176–190.
- [30] E. Zitzler, L. Thiele, Multiobjective optimization using evolutionary algorithms - a comparative case study, in: *Proc. Conf. Parallel Probl. Solving Nat.*, 1998, pp. 292–304.
- [31] T. Chen, M. Li, The weights can be harmful: Pareto search versus weighted search in multi-objective search-based software engineering, *ACM Trans. Softw. Eng. Methodol.* (2022).