Genetic Algorithm for the Determination of Linear Viscoelastic Relaxation Spectrum from Experimental Data

F. J. Monaco, A. C. B. Delbem, F. M. Federson

Department of Computer Sciences, ICMC-University of São Paulo, São Carlos, SP13560-970, Brazil

Received 10 July 2007; accepted 29 April 2008
DOI 10.1002/app.29496
Published online 16 March 2009 in Wiley InterScience (www.interscience.wiley.com).

ABSTRACT: Conventional procedures employed in the modeling of viscoelastic properties of polymer rely on the determination of the polymer's discrete relaxation spectrum from experimentally obtained data. In the past decades, several analytical regression techniques have been proposed to determine an explicit equation which describes the measured spectra. With a diverse approach, the procedure herein introduced constitutes a simulation-based computational optimization technique based on non-deterministic search method arisen from the field of evolutionary computation. Instead of comparing numerical results, this purpose of this paper is to highlight some subtle differences between both strategies and focus on what properties of the exploited technique emerge as new possibilities for the field. In oder to illustrate this, essayed cases show how the employed technique can outperform conventional approaches in terms of fitting quality. Moreover, in some instances, it produces equivalent results with much fewer fitting parameters, which is convenient for computational simulation applications. The problem formulation and the rationale of the highlighted method are herein discussed and constitute the main intended contribution. © 2009 Wiley Periodicals, Inc. J Appl Polym Sci 113: 122–135, 2009

Key words: genetic algorithm; parameter estimation; system modeling

INTRODUCTION

Viscoelasticity is a property of materials such as amorphous polymers, semicrystalline polymers, and biopolymers. Polymers' behavior for infinitesimal strains can be modeled by consititutive equations linking the stresses to the strains in the material. This behavior is time-dependent and can be described as follows:

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

where τ is the stress tensor, $G(t)$ is the linear relaxation modulus, and γ is the deformation tensor. Equation (1) shows the stress tensor, as function of linear relaxation modulus, and the rate of deformation tensor. In linear or nonlinear models of viscoelasticity, it is convenient to represent the modulus using a sum of exponential terms as follows:

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

where each λ_i is a relaxation time and G_i is its corresponding weight. Equation (2) shows the linear relaxation modulus as sum of exponentials. This modulus' representation requires a set of n pairs (G_i, λ_i), i.e., a discrete relaxation spectrum. This set can be determined from discrete experimental data.

The most commonly employed experiment to determine the relaxation spectrum of a polymer is small-amplitude oscillatory shear.[1] This experiment measures linear viscoelastic material's properties as storage modulus $G'(\omega_j)$ and loss modulus $G'(\omega_j)$ of the dynamic shear as functions of frequency, which may be expressed as follows:

$$G'(\omega) = \sum_{i=1}^{n} G_i \frac{(\omega_j\lambda_i)^2}{1 + (\omega_j\lambda_i)^2} \tag{3}$$

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

where ω_j is a particular frequency. The estimation process consists in properly setting pairs (G_i, λ_i) such that $G'(\omega_j)$ and $G''(\omega_j\ )$ from eqs. (3) and (4) best fit the experimental data. Equations (3) and (4) show the storage modulus of the dynamic shear and the loss modulus of the dynamic shear, respevtively.

It has already been observed that the spectrum estimation from experimental data is an ill-posed problem. Consequently, minor inaccuracy in the input data may produce major errors in the solution

Correspondence to: F. J. Monaco (monaco@icmc.usp.br).

Contract grant sponsors: São Paulo State Foundation for Research Support (FAPESP), National Council for Research and Development (CNPq), Coordination for Improvement of Higher Education Staff (CAPES).

© 2009 Wiley Periodicals, Inc.

spectra. This characteristic may lead to inadequate solutions with relatively high regression errors and spectra with unrealistic waved aspect[1,2] of the fitted curve because of the excessive degrees of freedom.

Aiming to overcome these drawbacks, various techniques have been proposed for the proper determination of the discrete relaxation spectrum. Orbey and Dealy[1] discussed and compared three methods: linear regression, linear regression with regularization, and nonlinear regression. The former two methods require n values to be preset such that they are equally distributed along a logarithmic scale; the one with regularization poses requirements on spectrum smoothness as an additional constraint independent of data. Ramkumar et al.[3] proposed another method using quadratic programming with regularization. The results obtained with this approach show that it is possible to estimate solutions with relatively smaller regression errors and reduced oscillation. Baumgaertel and Winter[4] (according to Ref. 1) proposed a nonlinear regression technique, which does not need λ coefficients to be preset, and is, therefore, capable of finding all the model's parameters.

Roths et al.[5] developed an edge-preserving regularization method that can properly resolve spectra with edges or at least large curvatures since the common regularization methods assumes the relaxation spectrum to be smooth. Gerlach and Matzenmiller[6] compared different numerical methods for the identification of the relaxation spectra. A gradient-based optimization algorithm is applied to solve the so-formulated nonlinear optimization problem with non-negative constraints on the unknown parameters. The same problem is also addressed using a three-step approach involving the determination of a qualified tendency function, a Tschebyscheff-approximation to this function, and a quadratic optimization method. Both approaches require no estimates of the relaxation times; however, they need apropriate starting values for the parameters to converge to stable minina. On the other hand, the windowing method of Emri and Tschoegl and the regularization technique based on singular value decomposition need information about the distribution of the relaxation times. The numerical methods requiring additional information provided a smoother distribution of the line spectrum.

Recent works have investigated other aspects of the determination of relaxation spectra. Zatloukal et al.[7] focused on the estimation of relaxation time spectrum from capillary steady shear and extensional data since capillary measurements are easy to obtain and this type of data is usually available for equipment designers. Malkin[8] investigated the advantages of fitting experimental data by using a continuous relaxation spectrum. Guzmán et al.[9] proposed a regularization-free method that can avoid the inversion of the integral equations via regularization by using a simple double-reptation model.

This article introduces a nonanalytical regression approach based on evolutionary algorithms for determination of relaxation spectra, which is suited to handle ill-posed problems and, likewise nonlinear regression methods, can also find all model's parameters. The field of evolutionary computation investigates nondeterministic search algorithms for complex optimization problems. These techniques can simultaneously iterate several solutions and combine the more prommising ones to generate a new improved set of solutions. EAs have been successfully applied to problems with nonlinear or even discontinuous objective functions, nonconvex objective function space, nonconvex search space, as well as ill-posed problems.

The results obtained by applying the proposed alternative to the determination of relaxation spectra of several polymers studied in the literature[1,2,3,10–12] show that it can produce curve adjustments with superior quality in terms of regression error and reduced waved aspect.

The remainder of this section offers a general conceptual background on evolutionary computation rationales, which are useful for both understanding the exploited technique and interpreting the reported results. "Methodology" section presents the employed methodology, and the experiments are described and commented in "Simulation experiments" section. Finally, "Discussion" section highlights the main conclusions drawn from the experiments.

Evolutionary computation

The field known as evolutionary computation encompasses methods for problem solving by simulating natural evolution process.[13,14] The employed computing techniques rely on iteratively applying random variations and subsequent tailored selection over a population (set) of prospective solution instances. Algorithms based on this method are known as evolutionary algorithms, out of which relevant examples include evolution strategies, evolutionary programming, and genetic algorithms (GAs), the latter being the approach employed in this work.

In-depth knowledge of either theoretical foundations or technical details concerning GA concepts are not required for properly understanding the content of this article. The following subsection is therefore meant only to provide a general overview of GA rationale and some background on how to interpret the obtained results. Its purpose is to introduce the contributions the method can offer to the focused problem domain.

Genetic algorithms

GAs were first introduced by John Holland and his colleagues at the University of Michigan in the 1960s. Conceptually, the method consists a simplified mimic of biological evolution by means of simulated genetic recombination and Darwinian natural selection. In terms of its metaphorical bioinspired jargon, the original or canonical GA accomplishes its duty starting from an initial population: a set of a few randomly chosen individuals. Each individual is actually a fixed-length binary string (i.e., an array of 0s and 1s) whose single bits parallel genes in an haploid chromosome. A carefully chosen encoding scheme is devised so that every bit-string pattern encodes one possible solution for the given problem.

The solution set, comprising the initial population is then evaluated by a fitness function, which expresses how suitably each individual is adapted to the environment, i.e., how "good" the solution is according to the employed criterion. Then, selection phase takes place as follows: a few specimens are chosen to compose a mating pool according to a selection rule, which grants the best-fitting individuals higher chances of being picked up. The subset formed is given the chance to reproduce, and the new individuals whereby produced are formed by the recombination of the chromosome of their parents, thus preserving information on good genes along generations. In addition to recombination, random mutation (arbitrary alteration) in every single gene of the descendants may also occur with a lower probability. With a bit of luck, chances are that some of the new individuals inherit the best genes from both their parents and then turn out to be even better in terms of the encoded solution. By repeatedly applying reproduction and selection operations over the ever evolving population, it is highly probable that after a reasonable number of generations there will be good solutions present. Recombination along with proportional selection tend to preserve good gene subsequences; mutation is useful to introduce new prospective gene sequences that would never be derived from current population solely by recombination operators.

It is important to highlight the fact that this simple mechanism of GAs does not require a detailed model of the problem to be known; all that is needed is a way to compare two solutions to decide which is better and thus guide the evolution process step-by-step toward its goal. This makes GAs a powerful general-purpose tool for optimization problems. Another noteworthy characteristics of a GA concerning conventional deterministic search methods is that it works with a population rather than with a single solution and therefore exhibits an implicit parallelism in the search process.

It is also worth to stress another important property of GAs, i.e., randomness. As it occurs in natural evolution, the fittest (best) individuals are given a higher probability of survival, rather than certainty—this prevents GAs from getting stuck in local optima as it is the case for conventional deterministic search methods. Thus, equilibrium between selection pressure on one hand and random variations on the other hand must be found. The stronger the former, the faster the evolution occurs, but higher is the probability of premature convergence (loss of global optimum). The stronger the latter, the faster is the exploration of the search space, but slower is the convergence to a global optimum.

This stochastic nature allows for an important assertion about (well designed) GAs as optimization tools: since a global optimum exists, there are good chances that its surrounding will eventually be approached, although it is not possible to known in advance how many generations it will take. As a nondeterministic method that relies on randomness-based operations, a GA can conduct the search domain exploration through different paths, depending on the sequence of values produced by its random number generator*; if the same deterministic pseudorandom function is utilized, the variations will then depend on the initial seed supplied to it. Therefore, for the interpretation of the results produced by the GA it is important to understand that, if the problem admits more than one solution within a given precision interval, successive runs of the algorithm with varying seeds may come out with different solutions of equivalent quality. This capacity of broad search-space exploration along several paths in parallel is useful for tackling the focused problem.

A number of variants of the canonical GA have been developed to make it applicable to other binary search space, and these include real-encoded[14] chromosome, which is the case for the highlighted problem.

METHODOLOGY

The regression problem tackled in this article can be posed in terms of a multiobjective optimization

*The GA algorithm performs several operations with basis on randomly chosen numbers. To produce aleatory output from deterministic computational systems, programs usually rely on pseudorandom generators: functions that compute different results upon successive calls, depending on the current value held in an internal state-variable that is automatically updated at every function evocation. The apparently aleatory sequence of numbers generated by consecutive functional calls must fulfill general proprieties of randomness such as homogeneous occurrence frequency and absence of repetition patterns. Distinct random sequences can be produced by initializing the algorithm internal state with an initial value often refereed to as random seed.

problem[15] in which one wishes to find a set C of n pairs (G_i, λ_i), $i = 1,2,\ldots n$, which makes the two parameterized functions of eqs. (3) and (4) best fit their respective experimental data set (the fitting qualities measured by the mean square errors, MSE). Alternatively, it is possible to convert it into a mono-objective problem by redefining the goal as the minimization of a linear combination of the MSE of each curve as in eq. (5):

$$M(G',G'',C)=\frac{K_{G'}\times \mathrm{MSE}(\mathrm{G}'_{\mathrm{C}})+\mathrm{K}_{G''}\times \mathrm{MSE}(\mathrm{G}''_{\mathrm{C}})}{K_{G'}+K_{G''}} \qquad (5)$$

where MSE(G'_{C}) and MSE(G''_{C}) are the MSE for $G'(\omega)$ and $G''(\omega)$ curve fittings, respectively, obtained by using the parameter set C, and $K_G{}'$ and $K_G{}''$ are weighting constants. Therefore, eq. (5) shows the mean square error.

Conceiving a GA-based approach involves the formulation of the problem in terms of its solution encoding, fitness function, search space, selection methods, recombination and mutation operators, and their corresponding rates.

As for the encoding scheme, the individual's chromosome may be represented by a sequence C of n genes, where every gene is a pair (G_i, λ_i), as illustrated in Figure 1. From eq. (5), a straightforward expression for the fitness function is $F(C)$, at the bottom of the figure (the minus sign makes the otherwise minimization problem into a maximization problem, only for intuitive consistency with the metaphor of the best-fitting survival): the lower the weighted MSE, the higher is the fitness.

The high-level description of the designed GA is in essence very simple, as depicted in Figure 2.

In the first step, the algorithm generates k individuals whose genes are randomly chosen within the search space in the range. Next, in Step 2, for each individual, the algorithm calculates G' and G'' values for the measured ω using the parameters encoded in the respective chromosome, and then compares the so-estimated value with the experimental data, assigning the corresponding specimen a fitness score.

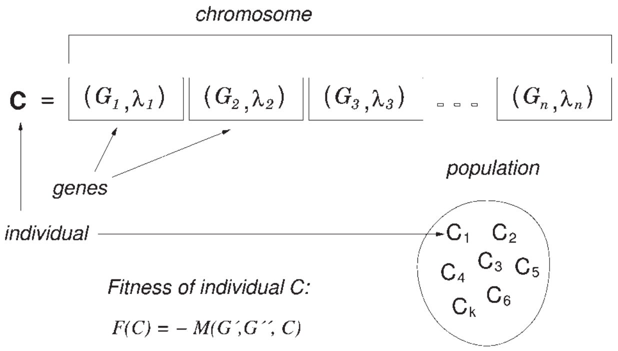

Figure 1 GA encoding scheme.

```
1. Create an initial population $P$ with $k$ individuals: $P = \{C_1, C_2, \ldots C_k\}$.
2. Evaluate the fitness of each individual: $F(C_j)$, $j = 1, 2 \ldots, k$
3. Select a subset of $s < k$ individuals to compose the mating pool.
4. Generate $d = k - s$ descendants from the parents in the mating pool.
5. Replace $d$ individuals from the population with the newly generated ones.
6. Repeat from step 2 until a good-enough individual has been found.
```

Figure 2 High-level outline of the implemented GA.

The composition of the mating pool in Step 3 is performed according to a method known as tournament of two: a selection scheme where the probability of choosing any individual is proportional to its position in the population fitness rank[14] (so that the best individuals are allowed higher chances to reproduce, thereby preserving positive characteristics along generations). Reproduction in Step 4 is also simple. Mating pairs are picked up from the pool in order, and for each couple, two descendents are generated by the recombination operator: a standard real-number crossover well described in the literature as the blend[15] method[†].

Next, every gene in the newly generated chromosome is subjected to mutation (small random deviation around the original value) with a low probability. As soon as the new d individuals have been created, d other individuals of the current population have to be discarded in Step 5. This is done again by the tournament of two, but this time the probability of selecting an individual is inversely proportional to its position in the population fitness rank (so that worse individuals are allowed a lower chance to survive). These individuals are then replaced with the just produced offspring to form the new generation. Steps 2–5 are repeated until either there is an individual whose fitness value is higher than a reference value r (i.e., the fitting MSE is less than the specified maximum) or the maximum allowed number of generations has been reached, in which case the algorithm ends up with the best-fitted individual up to that point. It is worth to note that an upper bound on the number of generation is only one of the reasonable iteration stop criteria. For the experiment, preliminary essays can be carried out to analyze the population convergence find a limit after which no further optimization takes place. An alternative is to choose a maximum admissible fitting error and cause the iterations to stop

[†] The blend operation has a pair of new individuals as outcome. The chromosome of one of the descendants is obtained by increasing or decreasing every gene (real number) of one of its parents by a small random value within a range that is proportional to the difference between the corresponding parents' genes. The second descendent is analogously generated from the chromosome of the other parent. Therefore, each one of the new individuals has its genes biased toward one of its parents.

TABLE I
GA Parameters Used in the Experiments

Parameter	Value
Population size	100
Substitution rate	0.3
Crossover rate	1.0
Mutation rate	0.01
Mutation range	0.1
Blend factor	0.5

when this quality is reached (this, however, requires knowing in advance that such fitting quality is possible from the experimental data; muticriteria stop conditions can be explored as well).

SIMULATION EXPERIMENTS

A program implementing the outlined procedure was written in C language using the standard GNU libc6 library, and compiled with gcc-3.3[16] on a GNU/Linux PC workstation. The software tool was named GenFitter.

The algorithm is controlled by a set of operational parameters that includes population size (how many individuals exist in the population), substitution rate (how many individuals are replaced at every generation), mutation rate (probability that a single gene suffers mutation), mutation range (random variation interval), crossover rate (probability of crossover occurrence), blend factor (which affects the blend crossover method[‡]), and initial random seed (for the random number generator). A series of preliminary empirical essays was carried out to determine a suitable set of parameters, which caused the algorithm to converge in a reasonable time. The values found are shown in Table I. For the weighted average of G' and G'' MSEs of the fitness function, as formulated in eq. (5), an equally balanced ponderation was adopted (a simple arithmetic average).

In addition to these parameters, the inputs of the program also include the desired fitness quality (specified precision, expressed as the maximum tolerated MSE), the maximum number of generations to be essayed, the data files containing the experimental measured values, and the number of pairs (i.e., the chromosome length) to be used in the estimation of $G'(\omega)$ and $G''(\omega)$.

To comply with most works in the literature, the average absolute deviation (AAD) measure,[3] shown in eq. (6), was utilized to compare curve-fitting quality between different regression methods, where the tilde-marked and nontilde-marked symbols mean experimental measurements and regression-estimated values M' and M'' are the number of available data for G' and G'', respectively.

$$\begin{aligned} \mathrm{AAD}(G') &= \frac{1}{M'} \times \sum_{j=1}^{M'} \left| \frac{(\tilde{G}'(\omega_j) - G'(\omega_j))}{\tilde{G}'(\omega_j)} \right| \\ \mathrm{AAD}(G'') &= \frac{1}{M''} \times \sum_{j=1}^{M''} \left| \frac{(\tilde{G}''(\omega_j) - G''(\omega_j))}{\tilde{G}''(\omega_j)} \right| \end{aligned} \tag{6}$$

The experiments described in the following subsections aim at the investigation of how GAs can contribute to rheology as a powerful technique for the determination of linear viscoelastic relaxation spectrum of polymers. The essay set includes a series of tests aimed at replicating the strategies exploited in previous works based on other techniques. The results obtained provided insights for further investigations on how either the number of parameters or the distribution along log λ-dimension affects regression quality.

Replication of previous works

By the way of evaluating the GA-based approach to determine the relaxation spectra of polymers, a set of essays reported in the literature was reproduced using GenFitter, and the results obtained were compared with those achieved by other techniques.

LDPE curve fitting with decimally spaced preset-λ coefficients

An important result of Laun's investigations[10] is that eight parameters with the second coefficient uniformly distributed over the log λ-dimension seem to be enough to suitably fit experimental data for the LDPE material. Following this indication, the first conducted experiment was meant to replicate the curve-fitting works reported in the cited reference, however using the introduced GenFitter to obtain the regression parameters. By presetting decimally spaced log λ-coefficients between −4 and +3, the method reported by the author produced the coefficients shown in Table II, whereas using the same preset λ-values and the experimental data obtained from the referred article, GenFitter found the coefficients shown in Table II.

With these estimated parameters and the experimental data set, from eq. (6), it is possible to calculate the ADD for G' and G'' fittings corresponding to the parameters listed by Laun: 0.0556 and 0.0591, respectively. These values are numerically

[‡] If c_i^n is the ith gene of the nth child and p_i^m is the ith gene of the mth parent, with $m,n \in [1,2]$ and $i \in$ [1, chromosome_length], then $c_i^1 = \mathrm{random}(p_i^1 - \alpha.\delta_i,\ p_i^1 + \alpha.\delta_i)$ and $c_i^2 = \mathrm{random}(p_i^2 - \alpha.\delta_i,\ p_i^2 + \alpha.\delta_i)$, where $\delta_i = |p_i^1 - p_i^2|$ and α is the blend factor.

TABLE II
Set of Eight Parameters Found To Characterize LDPE Material

G_i (Pa)	λ_i (s)
(a) As reported by Laun	
1.00×10^{-3}	$1.00 \times 10^{+3}$
1.80×10^{0}	$1.00 \times 10^{+2}$
1.89×10^{2}	$1.00 \times 10^{+1}$
9.80×10^{3}	1.00×10^{0}
2.67×10^{5}	1.00×10^{-1}
5.86×10^{6}	1.00×10^{-2}
9.48×10^{7}	1.00×10^{-3}
1.29×10^{9}	1.00×10^{-4}
(b) As found by GenFitter	
1.71×10^{0}	$1.00 \times 10^{+3}$
1.62×10^{2}	$1.00 \times 10^{+2}$
2.10×10^{3}	$1.00 \times 10^{+1}$
1.02×10^{4}	1.00×10^{0}
2.49×10^{4}	1.00×10^{-1}
5.84×10^{4}	1.00×10^{-2}
9.51×10^{4}	1.00×10^{-3}
1.71×10^{5}	1.00×10^{-4}

comparable with those associated with the slightly superior curve adjustment produced by GenFitter, whose corresponding AADs are 0.0532 and 0.0436. The graphic plot for the fitting curves generated by either technique is depicted in Figure 3, which shows that the oscillatory aspects exhibited by both fitted curves are also similar (note that oscillation is intrinsic to the experimental data).

Smoothening techniques for LDPE curve fitting

Results suggesting that the curve smoothness can be favored by decimally spaced log λ are provided by Nicholas and Tschoegl.[17] Aiming to obtain even smoother curves, more akin to the expected theoretical G' and G'' behaviors, Ramkumar et al.[3] exploited a regularization technique (see Evolutionary computation section), which requires the utilization of subdecimally spaced λ coefficients—0.2 on logarithmic scale—to obtain a suitable adjustment quality. The author used 41 uniformly distributed log λ values within the interval from −4 to +4. The same λ_is and set of experimental data were used to evaluate the GenFitter capacity.

Considering the errors of both regression experiments, Ramkumar's fitting ADD for G' and G'' are 0.01332 and 0.0098, respectively. Such outcome was sensibly outperformed by GenFitter, which produced an ADD of 0.0002 for G' and 0.0002 for G''. As seen in Figure 4, with regularly spaced distribution of log λs, GenFitter-adjusted curve does not sensibly lose in smoothness comparison with the other method, but gains substantially in precision.

With collections of few largely spaced measurement points, as the data set utilized in this experiment, regularization is an artifice that provides a smoothness improvement of fitted curves obtained by regression techniques. On the other hand, without such a constraint, the many degrees of freedom associated with the large number of parameters allow for GenFitter to find several ADD-equivalent fitting solutions, some of them are subjected to oscillation. If in addition to reduced error, smoothness is also sought after, then this additional goal must be somehow included in the algorithm's objective function.

For this purpose, if a smooth function behavior is assumed, a possible strategy that does not require modifications in GenFitter implementation consists in the utilization of an interpolation polynomial to complete interdata gaps with consistent extra points—the practical effect is therefore analogous to that of regularization. To compare the results, a second experiment was run with the same 41 preset-λs and a less sparse polynomial-generated data set

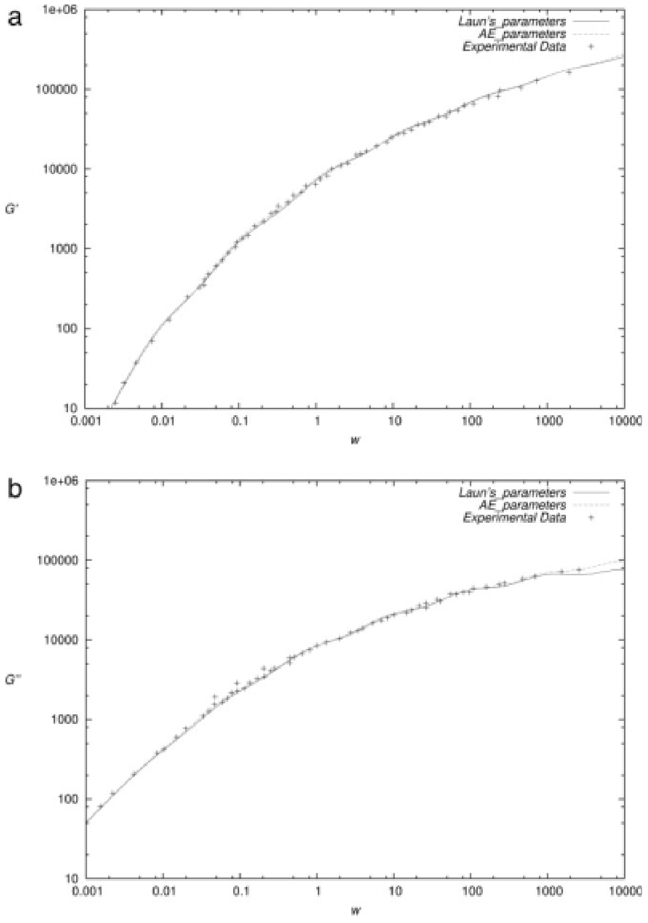

Figure 3 Curve-fitting plot with eight preset-λ coefficients for LDPE. (a) G' fitting by Laun and GenFitter and (b) G'' fitting by Laun and GenFitter.

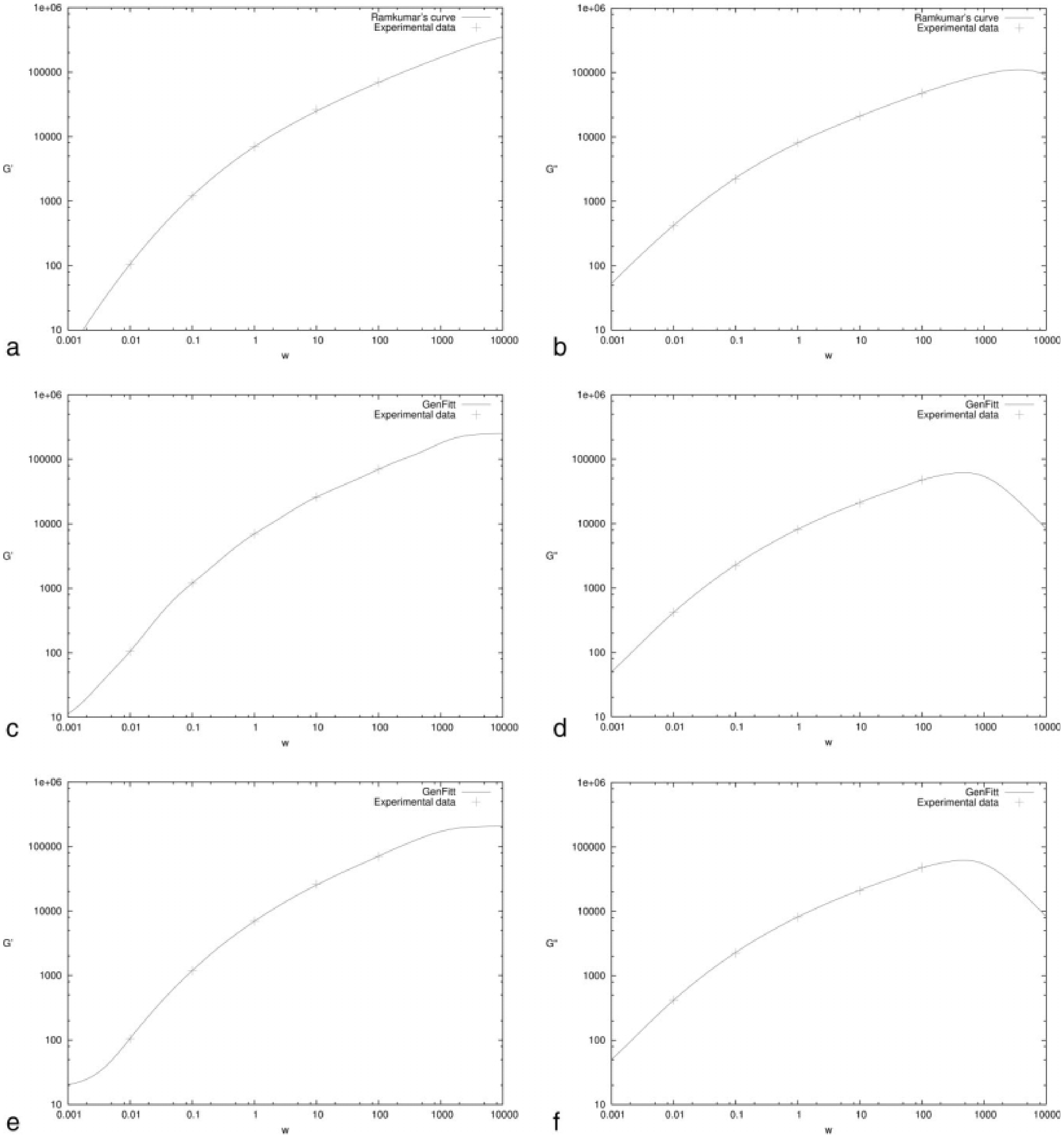

Figure 4 Curve plot for 41 subdecimally spaced preset parameters. (a) G' by Ramkumar; (b) G'' by Ramkumar; (c) G' by GenFitter; (d) G'' by GenFitter; (e) G' by GenFitter with smoothening; and (f) G'' by GenFitter with smoothening.

within the same range determined by Ramkumar. With these parameters, the AAD of G'- and G''-fitted curves with respect to the original experimental data were 0.01480 and 0.01226, respectively. Figure 4(c) shows the corresponding curve.

When the experiment was re-run with Laun's experimental data set, with eight parameters, the polynomial-smoothening technique proved also to be useful for a different purpose: to filter out measurement imprecision contained in the experimental data set. This allowed GenFitter obtain fitting curves with AAD 0.0548 and 0.04887 for G' and G'', respectively. Comparing these values with those previously obtained with raw data, it is possible to observe that precision was not sensibly affected and smoothness was improved, as illustrated in Figure 5.

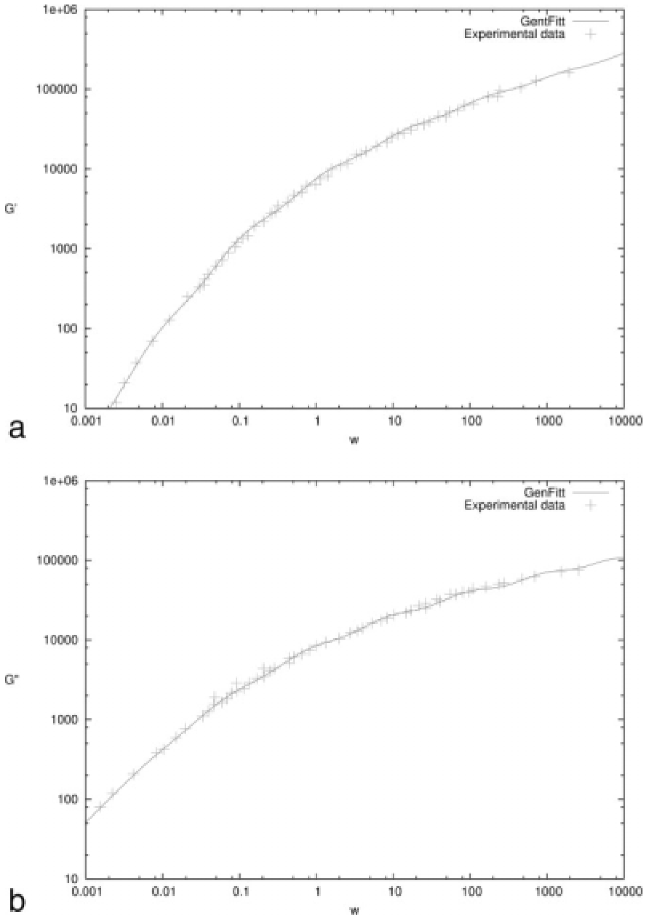

Figure 5 Plot for eight decimally spaced preset with smoothening technique. (a) G' by GenFitter with smoothening and (b) G'' by GenFitter with smoothening.

Nonregularly spaced log λ coefficients

Ramkumar et al. have shown that, in fitting G' and G'' from experimental data, smaller uniform spacing in log λ-dimension tends to favor smoothness and precision of the adjusted curve. As a consequence, the number of parameter pairs had to be increased to cover the whole spectrum. A larger number of parameters, however, is not convenient for the purpose of rheological simulation, inasmuch as it considerably increases computational load and therefore the processing time.

Considering there is no physical rationale asserting that regular spacing on log λ is required, the experiments reported in this section aimed to verify whether better results could be obtained with less coefficient pairs if this restriction were relaxed. By allowing the GA to freely explore the search space, if such a better solution exists, chances are that it will be approached during the evolution process. In a way of carrying out this experiment, GenFitter was modified so as to remove the constraint on λ values, which now are not preset.

TABLE III
AAD for LDPE Curve for 8 and 41 Pairs with Different Techniques

Method	G'	G''
(a) Laun's data [12] for LDPE		
Laun 8 pairs preset λ	0.0556	0.0591
GenFiffter 8 pairs free λ	0.0431	0.0410
(b) Zosel's data [18] for LDPE		
Ramkumar 41 pairs preset λ	0.0133	0.0098
GenFiffter 41 pairs free λ	0.0001	0.0001

The simulation for the cases of eight pairs (Laun) and 41 pairs (Ramkumar) was replicated, and the results obtained were compared with the previous ones in Table III. As it can be inferred, for both Laun's and Ramkumar's data, GenFitter is capable of either equaling or overcoming preset-λ experiments.

On the number of free-λ parameters

As it is observed from the performed experiments, setting λ free seems to lead to better solutions, since the GA is allowed more freedom to explore the search space. Would this advantage permit GenFitter to find solutions with fewer parameters with equivalent or improved quality than those previously found with either 8 or 41 pairs? What is the relation between the number of regression parameters and the fitting quality? To answer these questions, the simulation was re-run several times, varying the number of parameter pairs from 1 to 16. The corresponding regression errors are shown in Table IV.

TABLE IV
Free-λ Regression Errors for 1–16 Parameter Pairs (LDPE)

Number of pairs	Error for G'	Error for G''
1	4.0144	3.5398
2	0.5804	0.5434
3	0.3161	0.2609
4	0.1601	0.1439
5	0.0879	0.0790
6	0.0558	0.0481
7	0.0461	0.0450
8	0.0431	0.0410
9	0.0446	0.0393
10	0.0431	0.0410
11	0.0433	0.0410
12	0.0432	0.0410
13	0.0436	0.0387
14	0.0435	0.0401
15	0.0435	0.0401
16	0.0433	0.0392

As it can be observed, for the particular experimental data set (LDPE, Laun's listing), the employed technique (free-λ, raw data) allows no effective gain in precision beyond eight parameters, as it is also illustrated in Figure 6.

Empirical observations

This section outlines a few empirical observations that came out during the analysis of the results obtained from the described experiments.

On the search of space dimension

Figure 7 shows the relaxation spectra of LDPE (Laun's data) found by GenFitter in the free-λ experiment for different numbers of parameters. It is possible to notice that the larger the number of pairs, the larger is the range of λ_is found. For instance, the 10-pair set contains one element (e.g., the leftmost upper point in the referred graphic) outside the usual range of log λ adopted in the literature, which is usually between −4 and +4. This seems to comply with Tshoegl's suggestion that an adequate distribution of λ_is (but not necessarily in strict equally spaced pattern, as shown by GenFitter[§]) tends to smooth the curve. An inquiry has arisen from this observation. Are there better fittings, maybe with a reduced number of parameters, in an even broader search space? Because of the technical limitations, the current algorithm implementation is not suited for this case; in future works, the authors intend to address this issue.

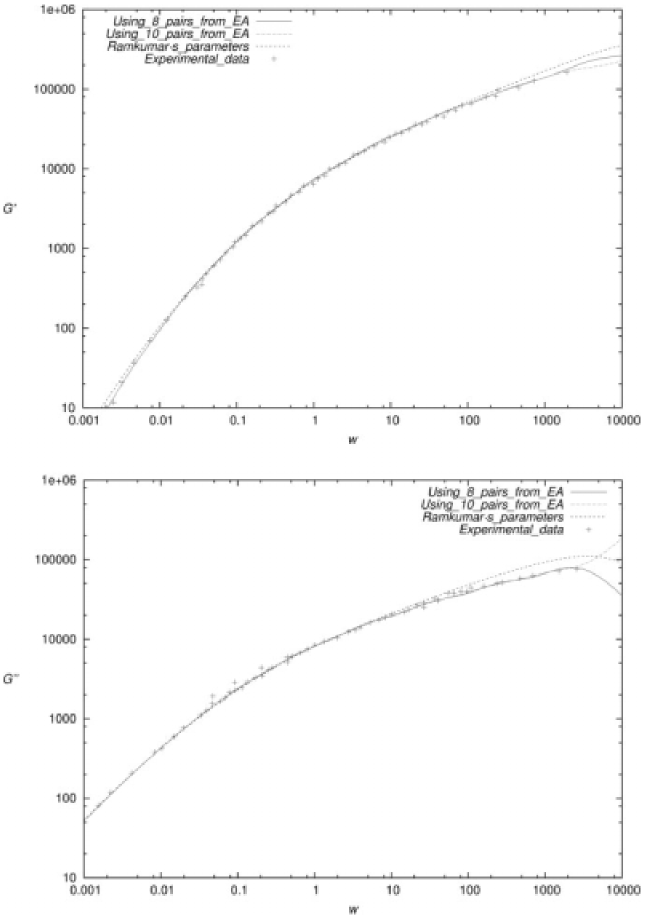

Figure 6 GenFitter-λ LDPE curve adjustment for 8 and 10 pairs. (a) G' fitting and (b) G'' fitting.

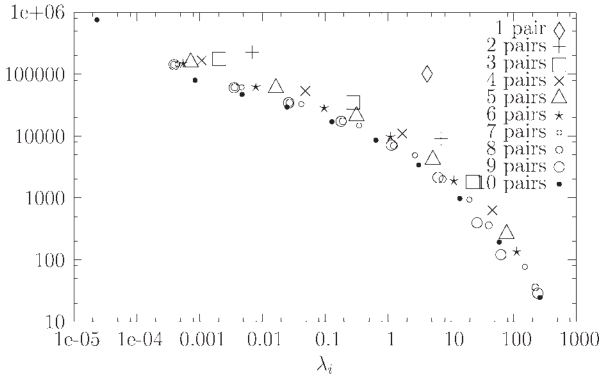

Figure 7 Relaxation spectra using different number of parameter pairs.

On the number of parameters

For a large number of parameters, when the algorithm is run several times with varying random seeds, different solutions (pair sets) are found with the same quality. This clearly indicates the presence of too many degrees of freedom, that is, some pairs are in excess for the given search stop condition (specified maximum MSE). To corroborate this result, Figure 8(a) shows the sets of both 41- and 10-coefficient pairs found by the free-λ GA runs (with smoothness). Several λ_is of the 41-pair solution set are approximately one or more decades below the λ_is estimated for the 10-pair parameter set, thus having less significance for the calculation of G' and G'' [see eqs. (1) and (2)]. Figure 8(b) highlights the selection of numerically significant parameters from the 41-pair set.

A comparison of G' and G'' plot with either the whole 41-pair set or the corresponding selected 10-pair subset is illustrated in Figure 9, in which it is evident that the removed extra pairs are of minor effect.

[§] The set of parameters found for varying number of free-λ parameters found by GenFitter revels that although not strictly equally spaced, the estimated coefficients indeed end up being not too heterogeneously distributed along its axis, thereby not in opposition to the rationale of Tschoegl's proposition.

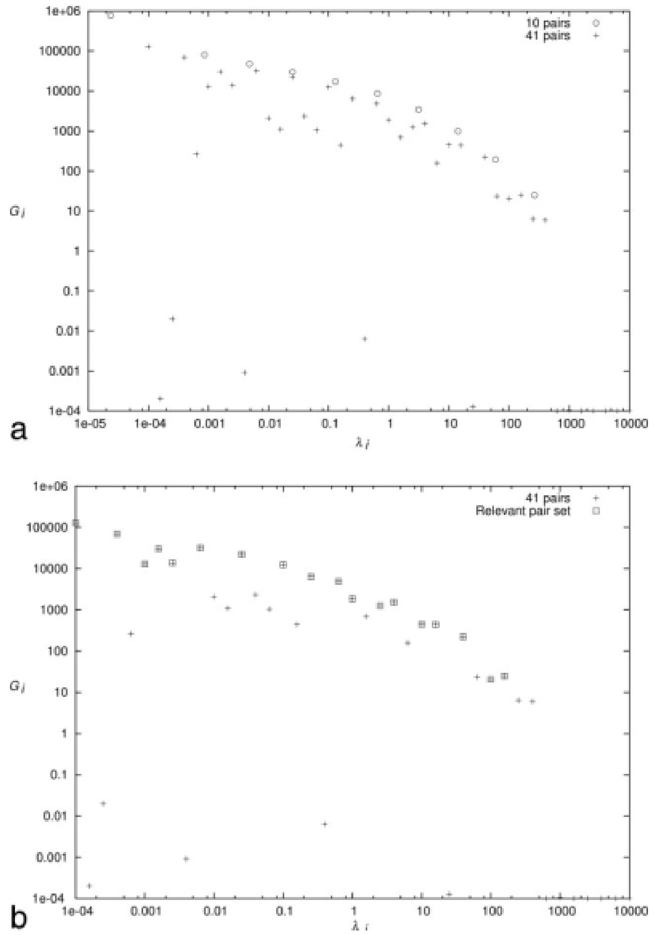

Figure 8 Comparison of the significant 41- and 10-parameter solution sets. (a) All the 41-pair and 10-pair solutions and (b) selected pairs from the 41-pair set.

From this behavior it is possible to infer that given a desired maximum fitting error, there exists a minimum number of pairs needed to properly represent the solution. What is this number? All the experiments with the GA were run for up to 100,000 generations only, so they allow no conclusive answer from the mere obtained numbers. However, a preliminary analysis of the algorithm convergence suggests that the values found are meaningful.

On the multiple solutions

Referring again to Figure 7, the distribution of log λ brings out a visual relationship between both parameters, which appear to be as disposed along a curve. When running the experiment several times, an interesting feature was then noticed: as the number of pairs increased, there were more and more points out of the observed curve. Besides, when the experiment was run with different random seeds, very frequently one of the two behaviors was observed. The points tended either to be close to the referred curve forming a cloud around it or they tend to be moved away from it, reaching the borders of the search space. The more the points added, the more evident becomes such phenomenon. Reasoning upon evolutionary computation rationales, it is interesting to conjecture about what takes place and why is happening. A sensible hypothesis is that, given a number n of pairs that is larger than what is necessary to fit the curve with the MSE below the specified upper bound, GenFitter has two ways of getting rid of the excess. One way is approaching pairs. It is easy to see from eqs. (3) and (4) that two pairs with λ_is next to each other may be replaced by a single pair with the former G_is summed up. The other way is by maximizing λ. Since both ways lead to low-error solutions, they tend to appear in the population after a number of generations. Determining the minimum number of coefficient pairs needed to properly fit the curve is useful not only for rheological simulation purposes but also to speed up the GA convergence.

The visual inspection of Figures 7 and 8 shows and apparent dependence of G_i upon λ_i, from which it is arguable to conjecture whether both parameters might be related by a deterministic function. If such hypothesis is eventually confirmed by further studies, then interesting contributions to the theory of viscoelasticity will be on the way. An immediate consequence would be the reduction of the number of parameters of eqs. (3)) and (4) from n to $n/2$ as investigated in Ref. 8. This article, however, does not propose to address this issue in depth; it is the subject of ongoing research by the authors.

Fitting G' and G'' for other materials

To verify the generality of the evolutionary computation approach for the focused problem, GenFitter

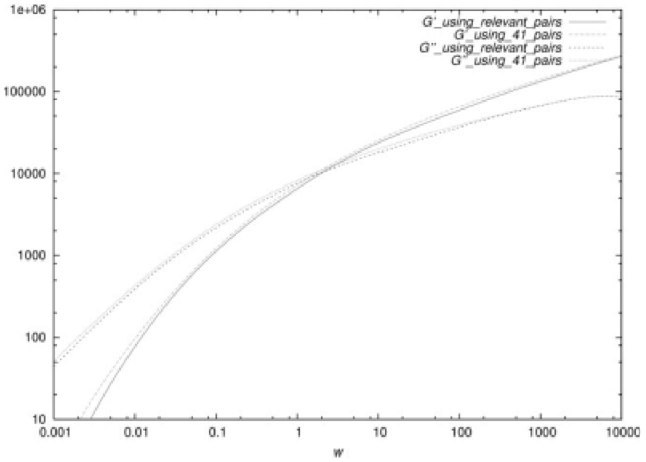

Figure 9 Comparison of G' and G'' plot with either all of part of the 41-pair set.

TABLE V
ADD by Different Methods to Fit G' and G'' for Diverse Materials

Method	n. par.	G'	G''	Average ADD
(a) Fitting results for polybutadiene				
Laun[5]		16.6	13.7	15.0
Baungaertel and Winter[4]		1.18	1.26	1.22
Honerkamp and Weese[2]		0.76	0.94	0.85
Ramkumar et al.[3]		0.018	0.011	0.015
GenFitter (minimum to outperform)	4	0.009	0.010	0.0095
GenFitter (maximum improvement)	8	0.007	0.007	0.007
(b) Fitting results for HDPE				
Laun[10]		4.82	2.73	3.75
Baungaertel and Winter[4]		1.33	1.12	1.23
HonerKamp and Weese [2]		0.85	1.26	1.05
Ramkumar et al. [3]		0.041	0.020	0.031
GenFitter (minimum to outperform)	6	0.022	0.014	0.018
GenFitter (maximum improvement)	10	0.017	0.017	0.017
(c) Fitting results for model data of Honerkamp and Weese[2]				
Laun[5]		6.89	10.16	8.57
Baungaertel and Winter[4]		5.12	5.35	5.24
HonerKamp and Weese[2]		3.00	4.10	3.55
Ramkumar et al.[3]		0.007	1.18	0.60
GenFitter (minimum to outperform)	7	0.029	0.034	0.32
GenFitter (maximum improvement)	10	0.029	0.039	0.34

employed to obtain the regression parameters for other polymers of importance in the literature of the area. In these experiments, the free-λ version was utilized, and since the experimental data do not present apparent unrealistic oscillation introduced by measurement error, no smoothening technique was employed. The results found and the best fitness available in the literature are presented in Table V. For each method, the table lists the number of adjustment parameters utilized; particularly for GenFitter, the list includes the minimum number of pairs needed to equal or outperform other methods, as well as the maximum number of such parameters that yields a better result, beyond which no significative improvement was obtained. As it can be observed, GenFitter results are notieceably positive, specially considering the lower number of parameters required when compared with other techniques.

Also, for the purpose of comparison, the fourth column in each table brings the average ADD of both previous columns. Since the optimization goal of GenFitter is the minimization of the average ADD, these values shows that even when other methods have found slightly inferior errors for either G' or G'' fittings, the GA was able to outperform all the others according to its implied objective functions, as in eq. (5). Should some level of priority be assinged to either fitting, this information can be encoded in the objective function of eq. (5), which can even use w-dependent ponderation to enforce better ajustment in different parts of the spectrum of each curve.

DISCUSSIONS

No investigative technique can be properly appreciated with respect to its realistic significance without sound pragmatic assessment, which take into account the peculiarities of the implied physical phenomena, conceptual principles, and methodological constraints. Without losing sight of this principle, it is worthy to remark that this article is neither meant to present a completely specified procedure for the determination of polymer relaxation spectrum nor intended as guide encompassing experimental aspects of parameter tuning or other method adjustments for practical usage. Rather, it focuses on exploring the characteristics of the proposed technique itself and its potential for the handling of the addressed problem domain. That is, there are sensible concerns regarding the practical usefulness of our proposed contribution, which deserve special annotations.

As a first remark, the end objective of the problem domain tacked in this work is to be able to predict time-dependent mechanical behavior of viscoelastic polymers by determining a discrete relaxation spectrum (G_i, λ_i). Naturally, one would like this to represent the behavior of a given phenomenon as it physically develops in the real world. In addition to the estimation of the error introduced by inherent imperfections of experimental procedure, the assessment of the obtained results must also take into account the available domain-specific knowledge about the system's expected properties. That is why, although the capability of the proposed technique to estimate fitting parameters for the studied curve was taken as a metric for the method evaluation, this does not mean that the sole criterion used for the success of introduced method can be how well it is able to fit the original data. Indeed, it is well known that spectra able to fit the data quite well can be pathological when compared with realistic continuous spectra, like, for instance, yielding negative relaxation strengths, which is quite unphysical.

In this study, approaching the example as a curve-fitting problem was one of the possible strategies to highlight the fact that nondeterministic simulation-based methods, upon which evolutionary computation is founded, inherently constitute general-purpose search techniques considerably independent of an in-depth knowledge about the object-system's model, in the sense that the algorithm itself does not incorporate any characteristic that might come from the specificities of the application domain; these can be incorporated into the search criteria.

Unlike conventional analytical methods, the knowledge of peculiarities of the system model (including both theoretical assumptions and experimental issues) is not intrinsically embedded into the algorithm itself, such as in the case of the curve smoothening by means of regularization techniques, or variable pondering constants aimed to selectively weight regression errors along different portions of the data set scale. Rather, GAs take this domain-specific knowledge as an input, in the form of a fitness function that encompasses whatever the criteria the users judges relevant to assess the quality of the solution he or she is expecting to find.

The fitness function can be as simple as a measure of linear regression error or as elaborate as a set of restrictions on the curve behavior according to realistic assumptions. As a practical consequence, the user can take our example algorithm exactly as it is and change only the fitness function to include any criteria he or she considers important to reflect both theoretical and experimental knowledge about the particular system under investigation.

Another relevant issue that may come about is when it is noted that GA method can yield different solutions of "equivalent quality" by varying the choice of the initial seed and there is no guidance as to how to select this. Indeed, because there exists an aleatory factor in the choices made by the GA during its execution, it immediately follows that successive runs of the algorithm will perform the search space exploration through different paths according to the particular random number sequence used, what in turn depends on the initial seed provided to the pseudo-random function. That is, if the algorithm execution converges to a given optimum, it can approach this same optimum (or other optima) from different directions. Therefore, successive runs can reach the termination condition in different points of the search space where the referred condition meets.

If this condition is, as in our case, that the best individual of the solution population is better than a given acceptable reference (low regression error), then every found solution reached in successive runs will be equivalent with respect to the evaluation criteria, that is, the fitness function. There is no meaningful way in which one solution can be considered better than the other because all meet the criteria we have established. If for some other criteria any solution may be considered better than others, then these criteria can be included in the fitness function and then those "better" solutions will naturally be privileged during evolution process. That is how we may include another restriction, for example, varying weights for regression errors in different extensions of the abscissa scale. Another related practical issue not addressed in our simulation experiments is the reliability of the inferred spectrum near the ends of the data range. Although this issue does not fit into the focus of this article, directions on this matter can be addressed straightforwardly: all that is needed is to provide a formula to numerically measure it and embed into the fitness function.

Summarizing the earlier discussion, it is worth to explain that GAs can be adapted and optimized in many ways. Relevant design decisions include the solution codification schema, recombination and mutation operators, initial population size, substitution rate at each generation, best-fit selection strategy, among other technical issues. These decisions have direct influence on the effectiveness of the results produced by the GA implementation, as they can determine how fast the search domain is explored, the tendency of premature convergence toward a local optimum, the robustness of the algorithm with respect to producing equivalent results in successive runs independently from the random seed utilized, and even the capability of reaching a reasonable solution.

The GA presented in this article, which we called GenFitter, was carefully tailored for the kind of problem under investigation. A good deal of background on evolutionary computation and exhaustive experimental work were employed to design a GA tunned to work considerably well for the viscoelastic relaxation model and its numerical constraints.

One more aspect to be noticed is that the simulation experiments were able to find suitable solutions for the problem with different number of parameter pairs, ranging from as low as 8 to as high as some tens. How is *n* selected? Obviously, if the number of parameters is allowed to be equal to the number of data points, all the data can be fitted with zero error. In such a case, one is fitting random error in the data. And using too many parameters leads to $G(t)$ plots with quite unrealistic features.

Actually, in the simulation experiments, *n* was not selected. The algorithm was run with several *n* values, and then the results were compared. What can be concluded from them is that there is both a lower and upper bound for the number of parameter pairs needed to fit the curve. Too small the value of *n*, the regression error is unacceptable; too large the value of *n*, there appears exceeding parameters with low significance in the final result. This was already expected from the theory. Nevertheless, the way the evolutionary computation simulation technique handles the issue is noteworthy.

Resorting to the metaphor of Neo-Darwinian evolution theory that grounds the GAs rationale, when the initial population is made of individuals with too many regression parameters, i.e., *n* is too large (computer-simulated), "natural" selection makes use of just a few of those parameters to produce well-fitted individuals. As it happens in natural evolution, genes that do not effectively contribute to improve the individual fitness do not tend to be preserved along generations. When too many parameters are used, the selective pressure naturally eliminates the useless genes by assigning them numerical values that do not interfere in the final result. It was even shown that the extra parameters fuzzily cloud around a well-defined curve of a small set of significant parameters (in biology terminology this effect is related to "genetic drift"). We also show that the final result is not modified if those excess of parameters is removed.

The objective of establishing the minimum number of pairs that will "properly" fit the curve is addressed by the "parsimonious" spectrum, which results from the method of Baumgärtel and Winter. However, they propose a systematical method to reduce the number of parameters. In the presented approach, the GAs find the minimum number of parameters by itself. It requires no mathematical or experimental assumptions and can work with free-λ sets. That is why *n* is not listed as an algorithm parameter in Table I.

To solve this ill-posed problem, some analytical methods add information to find a meaningful solution. The assumption of smoothness in $G(t)$ is the one most often used. This can be done by several means, including regularization techniques which often resort to some empirically determined constant. As an example only, this work uses a "polynomial smoothing technique," which has also been adopted in other problems of this kind. There are a few extra comments that can lay some light on this other feature of evolutionary computation when applied to the present problem domain.

As explained, to guide the evolution of prospective solutions toward an optimum, a criterion is needed to compare two solution instances and decide which is preferable. This criterion is given by the so-called fitness function. In our simulation experiment, we first adopted the regression error as the fitness criteria. That is, the evolution process pressed the solutions toward this goal, and this goal alone. Since the input data set includes experimental noise, lower regression errors are obtained when the curve fits such noise. To show that, if one judges conveniently, the selection criteria may be easily replaced (that is, it is not a part of the algorithm but a user-provided input), we extended the experiment to include a second criterion: to minimize regression error and at the same time to maximize the curve smoothness.

One of the possible methods to filter out random errors from the experimental data set is by means of an interpolation polynomial. This polynomial was then used to obtain a synthetic data set that arguably, upon the rationales of statistics, is a meaningful measure. Another way to achieve the same result would be to effectively modify the fitness function by making its value depend on both a measure of regression error and a measure of curve smoothness. This second approach would require a less trivial mathematical expression to quantify smoothness in numerical values, which could be significantly combined to the regression error values in the same equation. We found the first approach more suitable (compared with the regularization technique, for example, the interpolation polynomial is deterministically found, and does not depend on the choice of an arbitrary factor).

Finally, it can be observed that two data sets are used as the basis for comparing the simulation technique with other methods for the LDPE material: one is an old set obtained by Laun on a low-density polyethylene and the other set in fact is a model spectrum constructed by Ramkumar et al., all obtained in tabular data lists from the literature.[1,2,3,10–12,18] Actually, using modern up-to-date experimental data would be interesting for the purpose of producing a set of usable (G_i, λ_i) parameters. However, our main goal was to compare the results we could obtain from our GA-based technique with the ones reported by classical works of the literature. Since many authors have compared their methods with those by Laun and Rankumar, we considered interesting to base our evaluation on the same set of input data.

CONCLUSIONS

Out of the outlined discussion, it is possible to draw some conclusions. The first and foremost is that the proposed GA approach for the highlighted fitting problem succeeds in replicating the results obtained by conventional deterministic methods reported in the literature. Furthermore, when the method was evaluated for LDPE material and compared with the fitting techniques reported by Laun and Ramkumar, it was able to produce either equivalent or better solutions with less parameters.

Some conventional regression methods already used to solve the present fitting problem require one of the parameters, either G_i or λ_i, to be fixed *a priori*. In general, equally spaced values are chosen in the log λ axis, and then corresponding G values are calculated by deterministic procedures. On the other hand, this is not a requirement for the presented GenFitter: both G_i and λ_i are set free so that the GA can test any combination. The results show that there exist solutions with unequally spaced log λ that are better than those found under the restriction of preset λs. GenFitter also empirically determine the minimum number of parameter pairs required for an adequate adjustment of G' and G'' as performed in Refs. 1 and 4.

The proposed approach was evaluated for polymers studied in other works,[1–3,5–7] and the results obtained show that it can lead to solutions with relatively smaller regression error and reduced oscillation. Compared with linear regression, it does not require regularization and can suitably handle ill-posed problems. When explicitly required, smoothness could be included as a criterion by prefiltering experimental data by a interpolation polynomial—essays have shown that the results obtained are equivalent to regularization; although the herein introduced methodology does not rely on adjusted factors, which are independent of the data themselves.

The successful use of the proposed evolutionary computation method allows claiming that, despite their conceptual simplicity and easy implementation, GAs constitute a powerful tool for regression and multiobjective problems and can prove to be very useful for the determination of linear viscoelastic relaxation spectra.

References

1. Orbey, N.; Dealy, J. M. J Rheol 1991, 35, 1035.
2. Honerkamp, J.; Weese, J. Macromolecules 1989, 22, 4372.
3. Ramkumar, D. H. S.; Caruthers, J. M.; Mavridis, H.; Shroff, R. J Appl Polym Sci 1997, 64, 2177.
4. Baumgaertel, M.; Winter, H. Rheol Acta 1989, 28, 511.
5. Roths, T.; Maier, D.; Friedrich, C.; Marth, M.; Honerkamp, J. Rheol Acta 2000, 39, 163.
6. Gerlarch, S.; Matzenmiller, A. Int J Numer Methods Eng 2005, 63, 428.
7. Zatloukal, M.; Vlcek, J.; Tzoganakis, C.; Dobbie, T. Adv Polym Technol 2000, 19, 277.
8. Malkin, A. Y. Int J Appl Mech Eng 2006, 11, 235.
9. Guzmán, J. D.; Schieber, J. D.; Pollard, R. Rheol Acta 2005, 44, 342.
10. Laun, H. M. Rheol Acta 1978, 17, 1.
11. Meissner, M. Rheol Acta 1971, 10, 230.
12. Meissner, M. Rheol Acta 1971, 14, 471.
13. Goldberg, D. E. Genetic Algorithms in Search, Optimization, and Machine Learning; Addison-Wesley: Massachusetts, 1989.
14. Baeck, T.; Fogel, D.; Michalewicz, Z. Handbook of Evolutionary Computation; Oxford University Press and Institute of Physics Publishing: New York and Bristol, 1997.
15. Deb, K. Multi-Objective Optimization Using Evolutionary Algorithms; Wiley: New York, 2001.
16. The GNU Operating System. Available at: http://www.gnu.org, 2005.
17. Tschoegl, N. W. The Phenomenological Theory of Linear Viscoelastic Behavior; Springer-Verlag: Berlin, 1989.
18. Meissner, J. Pure Appl Chem 1975, 42, 553.