Journal of Macromolecular Science, Part A
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713597274

Neural Networks and Genetic Algorithms Used for Modeling and Optimization of the Siloxane-Siloxane Copolymers Synthesis

Silvia Curteanu[a]; Maria Cazacu[b]
[a] Department of Chemical Engineering, "Gh. Asachi" Techical University, Iasi, Romania [b] "Petru Poni" Institute of Macromolecular Chemistry, Iasi, Romania

To cite this Article Curteanu, Silvia and Cazacu, Maria(2008) 'Neural Networks and Genetic Algorithms Used for Modeling and Optimization of the Siloxane-Siloxane Copolymers Synthesis', Journal of Macromolecular Science, Part A, 45: 1, 23 — 36

To link to this Article: DOI: 10.1080/10601320701681896

URL: http://dx.doi.org/10.1080/10601320701681896

Journal of Macromolecular Science®, Part A: Pure and Applied Chemistry (2008) **40**, 23–36
Copyright © Taylor & Francis Group, LLC
ISSN: 1060-1325 print/1520-5738 online
DOI: 10.1080/10601320701681896

Neural Networks and Genetic Algorithms Used for Modeling and Optimization of the Siloxane-Siloxane Copolymers Synthesis

SILVIA CURTEANU[1] and MARIA CAZACU[2]

[1]*Department of Chemical Engineering, "Gh. Asachi" Techical University, Iasi, Romania*
[2]*"Petru Poni" Institute of Macromolecular Chemistry, Iasi, Romania*

Received May, 2007, Accepted June, 2007

This paper presents the use of neural networks and genetic algorithms as tools for modeling and optimization applied to a complex polymerization process–synthesis of statistical dimethyl-methylvinylsiloxane copolymers. A feed forward neural network models the dependence between the conversion of monomers and copolymer composition (output variables) and working conditions (temperature, reaction time, amount of catalyst and initial composition of monomers–input variables). The training and validation data sets are gathered by ring-opening copolymerization of the octamethylcyclotetrasiloxane (D_4) with 1,3,5,7-tetravinyl-1,3,5,7-tetramethylcyclotetrasiloxane (D_4^V), with a cation exchange (styrene-divinylbenzene copolymer containing sulfonic groups) as a catalyst, in the absence of solvent. This model is included into an optimization procedure based on a scalar objective function and solved with a simple genetic algorithm. The genetic algorithm computes the optimal values for the control variables and for the weight coefficients attached to the individual objectives. An inverse neural network modeling, that is the identification of reaction conditions leading to a desired value for copolymer composition, is presented as particular variant of optimization. The genetic algorithm and neural networks prove to be good and accessible tools for solving an optimization problem performed with a multi-objective scalar function and provide important information for the experimental practice.

Keywords: neural networks; direct and inverse modeling; genetic algorithms; optimization; siloxane copolymer; polydimethylmethylvinylsiloxane

1 Introduction

The implementation of mechanistic models that rely on many processes from the chemical industry, as well as empirical correlations, involves a great deal of mathematical difficulties and, in many instances, lacks accuracy. Neuron-based modeling can be used confidently as a substitute for such situations. This is due to the favorable features entailed in their use. Among these features are: simplicity, fault and noise tolerance, plasticity property (can retain its prediction efficiency even after the removal or damage of some of its neurons), black box modeling methodology, capability to adapt to process changes (1).

Artificial neural networks (ANN) have been widely used for many different industrial areas such as control, prediction, pattern recognition, classification, speech and vision. ANNs have been trained to solve nonlinear and complex problems that are not exactly modeled mathematically or for which insufficient knowledge is available. ANNs extract the desired information using the input-output data and can be used to deal with the problems with incomplete and imprecise input data.

The polymerization processes are typical examples for neural network based modeling because of a series of difficulties such as the complex reactions occurring simultaneously inside the reactor, the large number of kinetic parameters which are usually not easy to determine, as well as the poor understanding of chemical and physical phenomena for mixtures involving polymers.

The open literature presents many attempts concerning neural network applications for polymerization processes: *direct modeling* with different types of neural networks (2–5), neural networks based *soft sensors* (6), *inferential modeling* (7, 8), *inverse neural network modeling* (5, 9, 10), *optimization* (11–14), *process control* (15–17). These types of applications are reviewed in our precedent work (18). Fernandes and Lona provide a brief tutorial on simple and practical procedures that can help in selecting and training neural networks and address complex cases where the application of neural networks has been successful in the field of polymerization (19).

Process optimization and control can have a significant strategic impact on polymer plant operability and economics.

Address correspondence to: Silvia Curteanu, Department of Chemical Engineering, "Gh. Asachi" Techical University, Bd. D. Mangeron No. 71A, 700050, Iasi, Romania. E-mail: silvia_curteanu@yahoo.com

Polymer production facilities face increasing pressures for production cost reductions and more stringent quality requirements (12).

Generally, the optimization of a polymerization process is multi-objective in nature, since it normally has several objectives, often conflicting and non-commensurable, that must be satisfied at the same time. Therefore, solving such a problem is accompanied by difficulties starting with the way of formulating the objective function and continuing with the choice of working procedure and selection of the results from more possible options. In the last several years, some research has been reported in the open literature on the optimization of polymerization reactors using multiple objective functions and constraints, and its use in on-line optimizing control.

A multi-objective function can be formulated using *weighted average approach* (scalar approach) (20, 21) or can be *a vector of objective functions* where all the objectives are treated simultaneously to find a set of all the solution (22, 23). The first approach allows simple algorithms to be used for solving the problem, but depends on the user's decision to specify weights to the different objectives based on good knowledge of the process. On the other hand, the user has the possibility to change the priorities of the partial objectives and solve the problem to get a number of different solutions depending on technological criteria.

In recent years, there is a growing interest in the optimization techniques based on evolutionary algorithms, particularly *genetic algorithms*. Because of their flexibility, ease of operation, minimal requirements and global perspective, these algorithms have been successfully used in a wide variety of multiobjective problems. Multi-objective optimization of the polymerization processes is an example of their applications (23, 24). These techniques do not need any initial guesses and converge, in most cases, to the global optimum even when there are several local optima present. In addition, genetic algorithms use information about the objective function and not its derivatives (such traditional optimization techniques), nor do they require any other auxiliary knowledge about the process.

Many types of genetic algorithms have been described and applied, the literature being extensively reviewed from this point of view in the papers of Coello (25), Deb (26) and Carlos (27), by marking the advantages and disadvantages of these types of algorithms in simple examples. Different types of genetic algorithms and their applications in chemical reaction engineering, including polymerization processes, have been pointed out in the reviews of Bhaskar (28) and Nandasana (29). The efficiency of optimization with vectorial objective functions in the frame of genetic algorithms or genetic programming (30) was proven in relevant practical applications.

Polysiloxanes, also named as silicones, are the most important inorganic polymers, the representative term being polydimethylsiloxane. Silicone polymers containing organic groups, other than methyl or specific organic function on the chain or at its ends, have opened new fields of applications which are a result of siloxane chemical reactivity, solubility, miscibility, paintability, lubricity, etc.

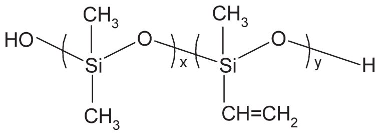
Dimethyl-methylvinylsiloxane copolymers are very important precursors for a post-functionalization, because the vinyl groups can easily be transformed into a variety of other functional groups. Block or statistical copolymers can be obtained depending on the chosen reaction pathway (31). The research developed in the past followed obtaining copolymers having tailored microstructure.

A facile and useful method for the synthesis of statistical dimethylmethylvinylsiloxane copolymers (Sch. 1) was chosen in this paper to apply the methods of artificial intelligence for modeling and optimization. That is, the ring-opening copolymerization of the octamethylcyclotetrasiloxane with 1,3,5,7-tetravinyl-1,3,5,7-tetramethylcyclotetrasiloxane in the presence of a solid acid as a catalyst and in the absence of solvent.

This paper presents the use of neural networks and genetic algorithms as tools for modeling and optimization applied to a complex polymerization process–synthesis of dimethylmethylvinylsiloxane copolymers. As mentioned above, the reactions for polysiloxane synthesis are very complex, with many reactions occurring concomitantly. The variation in time of the main parameters of the process (conversion and copolymer composition) was modeled by using direct neural network modeling. The inverse neural network modeling, consisting in the determination of reaction conditions that lead to pre-established properties (copolymer composition), is also performed in this paper.

A neural network model is then included into an optimizing control scheme, which uses a genetic algorithm solving technique and a multi-objective function in a scalar form. The partial objectives are the maximization of the reaction conversion and the achievement of a desired value for copolymer composition. The decision variables are reaction temperature, concentration of the catalyst, reaction time and initial composition. Our approach presents the advantage of computing the optimal values for the weights of the objectives within the genetic algorithm, along with the optimal values for decision variables.

The contribution of this paper refers mainly to the modeling and optimization capacities of the simple topologies and simple working strategies of neural networks and genetic algorithms, applied for the first time in the polysiloxanes reaction field. The procedure developed herein is quite general and it can easily be used for other optimization problems.



Sch. 1. General structure of the dimethylmethylvinylsiloxane copolymer.

2 Experimental

Statistical dimethylmethylvinylsiloxane copolymers are synthesized by ring-opening copolymerization of the octamethylcyclotetrasiloxane (D_4) with 1,3,5,7-tetravinyl-1,3,5,7-tetramethylcyclotetrasiloxane (D_4^V) using a cation exchange (styrene-divinylbenzene copolymer containing sulfonic groups) as a catalyst and working in the absence of solvent. Water is used as a co-catalyst and chain transfer agent (32–34).

In the presence of the strong acids, as well as bases, the Si-O bonds in both unstrained cyclosiloxanes and linear macromolecule (which have comparable energy) can be split, and a mixture of cyclic and linear polysiloxanes will be obtained, according to Sch. 2. The siloxane bonds are continuously broken and reformed until the reaction reaches a thermodynamic equilibrium.

The equilibrium position depends on the starting cycle size, the substituent nature, and also on the reaction conditions (concentration of cyclosiloxane units, solvent, initiating system, and temperature). Therefore, the reactions for polysiloxane synthesis are very complex, a series of ring-opening polymerization, polycondensation, depolymerization by cyclization and chain scrambling reactions occur at the same time, except in the case when the conditions for the kinetic control are created. It is of great interest to know the conditions in which the copolymers with desired compositions in maximum yields can be obtained.

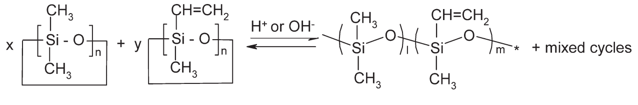
2.1 Materials

Octamethylcyclotetrasiloxane, $[(CH_3)SiO]_4$, D_4, supplied by Fluka AG with the following characteristics: b.p. = 175°C; $n_D^{20} = 1.396$; $d_4^{20} = 0.955$, purity >99% (GC).

1,3,5,7-tetravinyl-1,3,5,7-tetramethylcyclotetrasiloxane (D_4^V) (Fluka AG): b.p. 111/10 mm Hg; $n_D^{20} = 1.4343$; $d_4^{20} = 0.9892$, purity >99% (GC).

Purolite CT175-a macroporous styrene-divinylbenzene copolymer with sulfonic groups, supplied by Viromet,was utilized after conditioning, which consists of: washing with water, washing with a 4% NaOH solution, regeneration with a 4% HCl solution, washing with water and subsequent dehydration by azeotrope distillation with toluene and vacuum drying (110°C, 20 mm Hg).

The cation-exchanger has the following characteristics: active group $-SO_3H$; volumic exchange capacity-1.87 meq/ml; gravimetric exchange capacity-4.5 meq/g; specific surface-35 m^2/g.



Sch. 2. Reaction scheme for the obtaining dimethylmethylvinylsiloxane copolymer.

2.2 Techniques

^{1}H-NMR spectra were recorded on a JEOL C-60 HL Spectrometer using $CDCl_3$ as solvent, and the chemical shifts were reported to its residual signal.

The kinematic viscosity, η (cSt), was determined at 25°C, with an Ubbelohde Suspended Level Viscometer by using the relationship: $\eta = k \cdot t$, where k is the viscometer constant and t is the flow time for polymer, in seconds.

Gel permeation chromatographic analyses (GPC) were carried out on a PL-EMD 950 Evaporative Mass Detector instrument by using chloroform as an eluent after calibration with standard polystyrene samples.

2.3 Procedure

Cyclic monomers (D_4/D_4^V mixture) were loaded in a reaction vessel equipped with a thermometer, reflux condenser, and mechanical stirrer, then immersed in a thermostated bath at the pre-established temperature, according to the experimental program (a second order, rotable, composed, centered program). The cation-exchanger, Purolite CT-175, as a catalyst (2.5 wt.% reported to the reaction mixture) and water, as a co-catalyst (in a pre-established percent reported to the reaction mixture) were added when the desired temperature was attained in the reaction vessel. This is the initial moment of the reaction. The reaction mixture was stirred in these conditions for a certain time, according to the experimental program, after which the catalyst was removed by filtration. The reaction mixture was devolatilized by heating at 150°C/5 mm Hg. The remained polymer was weighted to calculate yield and analyzed (the determination of the composition) (34).

The copolymer composition was determined on the basis of ^{1}H-NMR spectra, as: $F_1 = [\text{-}(CH_3)_2SiO\text{-}]/\{[\text{-}(CH_3)_2SiO\text{-}] + [\text{-}(CH_3)(C_2H_3)SiO\text{-}]\} = (a - b)/(a + b)$, with a, b–the area of the signal corresponding to methyl and vinyl protons, respectively.

Experimental data are presented in Table 1.

2.4 Optimization Strategy

An optimization problem can be formulated in terms of the following elements: an accurate model of the process, a selected number of control variables, an objective function and a suitable numerical method for solving the specified optimization problem.

The optimization procedure includes a neural network model (NN–neural network), which can be represented:

$$\text{NN[Inputs: } t, T, C, f_1; \text{ Outputs: } x, F_1] \tag{1}$$

where t is the reaction time, T is the temperature of the copolymerization process, C–amount of catalyst, f_1–initial composition of the reaction mixture, x-reaction conversion, F_1–copolymer composition.

Table 1. Experimental data obtained for copolymerization of the octamethylcyclotetrasiloxane (D_4) with 1,3,5,7-tetravinyl-1,3,5,7-tetramethyl-cyclotetrasiloxane (D_4^V)

No. crt.	Reaction time t, [hr]	Reaction temperature T [°C]	Amount of catalyst C, [%-g]	Initial composition f_1[a]	Conversion x	Copolymer composition F_1[b]
1	0.75	45	2.5	0.3	4.44	0.536
2	1.75	45	2.5	0.3	8.28	0.535
3	0.75	75	2.5	0.3	17.82	0.458
4	1.75	75	2.5	0.3	41.50	0.439
5	0.75	45	4.5	0.3	4.46	0.550
6	1.75	45	4.5	0.3	12.60	0.503
7	0.75	75	4.5	0.3	37.96	0.476
8	1.75	75	4.5	0.3	55.68	0.411
9	0.75	45	2.5	0.7	10.12	0.863
10	1.75	45	2.5	0.7	20.00	0.848
11	0.75	45	2.5	0.7	34.74	0.835
12	1.75	75	2.5	0.7	56.31	0.799
13	0.75	45	4.5	0.7	19.86	0.878
14	1.75	45	4.5	0.7	44.96	0.846
15	0.75	75	4.5	0.7	48.98	0.813
16	1.75	75	4.5	0.7	68.61	0.777
17	0.25	60	3.5	0.5	6.22	0.752
18	2.25	60	3.5	0.5	49.18	0.641
19	1.25	30	3.5	0.5	8.44	0.747
20	1.25	90	3.5	0.5	61.70	0.591
21	1.25	60	1.5	0.5	11.71	0.721
22	1.25	60	5.5	0.5	44.78	0.654
23	1.25	60	3.5	0.1	15.29	0.225
24	1.25	60	3.5	0.9	46.18	0.956
25	1.25	60	3.5	0.5	34.20	0.681
26	1.25	60	3.5	0.5	30.10	0.684
27	1.25	60	3.5	0.5	30.84	0.701
28	1.25	60	3.5	0.5	33.62	0.693
29	1.25	60	3.5	0.5	30.41	0.690
30	1.25	60	3.5	0.5	32.53	0.687
31	1.25	60	3.5	0.5	31.42	0.684

[a] Feed molar ratio, $f_1 = [\text{-}(CH_3)_2SiO]_4/\{[\text{-}(CH_3)_2SiO\text{-}]_4 + [\text{-}(CH_3)(C_2H_3)SiO\text{-}]_4\}$.

[b] Copolymer composition, $F_1 = [\text{-}(CH_3)_2SiO\text{-}]/\{[\text{-}(CH_3)_2SiO\text{-}] + [\text{-}(CH_3)(C_2H_3)SiO\text{-}]\}$.

The control variable vector, u, has as components:

$$u = [t, T, C, f_1] \tag{2}$$

An admissible control input u^* should be formed in such a way that the performance index, J, defined by the following equation, are minimized:

$$J = w_x \cdot (1 - x_f) + w_{F1} \cdot \left(1 - \frac{F_{1f}}{F_{1d}}\right)^2 \tag{3}$$

subject to:

$$\begin{aligned} &0 \le x_f \le 1 \\ &20 \le T \le 110^\circ\mathrm{C} \\ &0.25 \le t \le 2.25\,\mathrm{h} \\ &0.5 \le C \le 5\% - g \\ &0 \le F_1 \le 1 \end{aligned} \tag{4}$$

In the Equation (3), J is the objective function to be minimized, w are weighting factors (w_x for conversion and w_{F1} for copolymer composition), x_f is the conversion at the end of the reaction, F_{1d}-desired value of the copolymer composition, F_{1f}-actual value corresponding to the final reaction time.

The objective function includes the maximization of monomer conversion, which leads to higher productivity. The constraint on reaction conversion forces the amount of unreacted monomers to be small, and hence, keeps post-reactor separation and recycling costs low. The endpoint requirement on F_1 leads to the production of copolymer having desired properties.

Constraints that are present in industrial reactors are very important to define the range of variation of parameters and to disregard possible solutions that might be interesting in a theoretical approach to the problem.

The optimization procedure includes a neural model and is solved with a genetic algorithm (GA). The fitness function of the GA is the scalar objective function (3). Figure 1 illustrates this optimization procedure. Genetic algorithm provides, after an iterative calculus, the optimal values for decision variables (t, T, C, f_1), which are the inputs for the neural network model and the weights for the objective function. With these inputs, the neural network computes the parameters x and F_1, and the last one will by compared with F_{1d}. If the two values are identical or the difference between them is very small, we can conclude that the task of the optimization, represented by minimum of the objective function, is achieved.

A good process model—accurate model with short simulation time—is a prerequisite for application in the optimal control strategy. A phenomenological model of a polymerization process is difficult to obtain and use, especially in complex applications such as optimal control of the process. The difficulties lie with the complex and numerous reactions and chemical species occurring simultaneously inside the reactor, the large number of kinetic parameters that are usually not easy to determine, as well as the poor understanding of chemical and physical phenomena for mixtures involving polymers. It was mentioned above that the reactions for polysiloxane synthesis are very complexes including a series of ring-opening polymerization, polycondensation, depolymerization by cyclization and chain scrambling reactions occurring in the same time. Consequently, the modeling with neural networks can overcome these difficulties due to a series of advantages: the possibility of applying this method to complex non-linear processes, the ease in obtaining and using neural models, the possibility of substituting experiments with predictions. Neural networks possess the ability to learn what happens in the process without actually modeling the physical and chemical laws that govern the system. Neural models need only input-output data (experimental or simulation data) so, their advantages are evident against the complexity of the computation.

The neural network modeling implies the following stages: collecting the training data by experiments, making up the training and testing data sets, developing the neural network topology, training and, finally, establishing the performance of the neural network model by comparing the network prediction to unseen (validation) data.

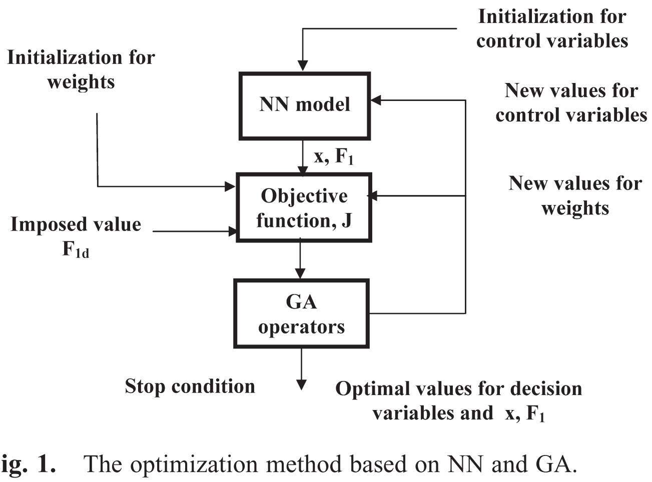

Initialization for control variables
Initialization for weights
NN model
New values for control variables
x, F1
Objective function, J
New values for weights
Imposed value F1d
GA operators
Stop condition
Optimal values for decision variables and x, F1

Fig. 1. The optimization method based on NN and GA.

In general, a neural network consists of processing neurons and information flow channels between the neurons, usually called, “interconnections”. Each processing neuron calculates the weighted sum of all interconnected signals from the previous layer plus a bias term and then generates an output through its activation transfer function. The adjustment of the neural network function to experimental data (learning process or training) is based on a non-linear regression procedure. Training is done by assigning random weights to each neuron, evaluating the output of the network and calculating the error between the output of the network and the known results by means of an error or objective function. If the error is large, the weights are adjusted and the process goes back to evaluate the output of the network. This cycle is repeated till the error is small or a stop criterion is satisfied (35).

The purpose of developing a neural model is to devise a set of formulae that captures the essential relationships in the data. These formulae are then applied to new sets of inputs to produce corresponding outputs. This is called generalization and represents subsequent phase after training-validation or testing phase. Since a neural network is a nonlinear optimization process made up of learning and testing phases, the initial data set must be split into two subsets: one for training and one for testing. A learning algorithm should lead to a good fit to the training samples and, simultaneously, to a network that has a good generalization capability. A network is said to generalize well when the input-output relationship, found by the network, is correct for input/output patterns of validation data, which were never used in training the network (unseen data).

The quality of the models strongly depends on the quality of the involved neural networks. It is well known that the construction of an efficient neural network is a function of many factors. The amount and appropriateness of the available training data is an important factor. Our experimental data correspond to this statement from both points of view: a considerable number of experiments were carried out and the chosen conditions cover the whole domain of interest. It must be mentioned that an experimentally composed centered program of second order was used.

A multi-layered, feed-forward, fully connected network is chosen in this paper for the copolymerization process modeling. Reasons for the use of this kind of neural network are the simplicity of its theory, ease of programming and good results and because this neural network is a universal function in the sense that if topology of the network is allowed to vary freely, it can take the shape of any broken curve (19). The topology of the network is developed by a trial and error method, following a balance between complexity and performance.

GAs are among the most widely used stochastic search algorithms and represent a promising alternative to gradient-based optimization techniques for different classes of problems. GA mimics the mechanics of natural selection and natural genetics to find the solution of optimization problems. Starting from an initial population (an initial set of solutions), generated on a random basis, new populations are created repetitively from the parent population until a solution is determined. Each solution in the population is called a chromosome (or individual) and represents a point in the search space. In a GA, a fitness value is assigned to each individual according to a problem-specific objective function. Generation by generation, the new individuals, called offspring, are created and survive with chromosome in the current population, called parents, to form a new population. Based upon the Darwinian theory of survival of the fittest, only individuals well fit for the environment (i.e., optimization problem) proceed to the next generation.

The chromosomes are evolved through successive iterations, called generations, by genetic operators: selection, crossover and mutation, until a stopping criterion is satisfied. In a GA, a set of solutions are analyzed and modified by genetic operations simultaneously, where selection operator can select some "good" solutions as seeds, a crossover operator can generate new solutions, hopefully retaining good features from parents, and a mutation operator can enhance diversity and provide a chance to escape from local optima (36, 37).

In this paper, a simple genetic algorithm (SGA) with real value encoding for the chromosomes was used. The initial population is generated randomly. Offspring is created by genetic operators and it is stored in a population pool that is a collection of offspring and their parents.

There are different methods for the selection phase; our paper uses rank selection which first ranks the population and then every chromosome receives fitness from this ranking. The individuals with higher fitness must have more chances to reproduce.

The *recombination* (*crossover*) has as main purpose, the recombination of the features of two randomly selected parents from the mating pool with the aim of producing better offspring. The variant of crossover used in this study presumes different points for all genes, that means the new individual will no longer be on the line segment that links its parents. The offspring will look more like one parent regarding a feature and less regarding another.

After recombination, offspring undergoes to *mutation*. Generally, the mutation refers to the creation of a new chromosome from one and only one individual with predefined probability. Mutation is used to produce small perturbations on chromosomes to promote diversity of the population. Our GA includes a variant of mutation named resetting. A gene value is reset to a random value in its search interval. The purpose is to refresh the search process, in a case when the genetic diversity of the population decreases (so no longer converges to the solution) or the algorithm has converged into a local optimum. Each gene is independently considered, and mutation gives it a new random value in the initialization interval. Only some genes change (possibly all, but unlikely).

After three operators are carried, the offspring is inserted into the population, *replacing* the parent chromosomes in which they were derived, producing a new generation. The best individual is copied directly into the new population (the elitism technique) and the rest of the individuals are replaced by the new generations. So, in order to keep the best solution, we have considered an elitism factor $f_e = 1$, that is the best individual is copied directly in the new generation. That ensures that the overall solution of the GA will not get worse.

The termination criterion determines when GA will stop. In other words, the genetic operations are repeated until a termination condition is met. In our implementation, we stop GA, if a maximum number of generations has been executed or reaches the pre-set number of generations without improvement in the last best solution.

Figure 2 presents the general flowchart of a simple genetic algorithm.

Therefore, the optimization procedure proposed in this paper is composed by two stages: modeling with neural networks and optimization based on genetic algorithms.

1) In our work, modeling the studied process assumes:
   - Establishing the neural network topology and its training with the well known back-propagation algorithm.
   - Validating the neural model with data not used in the training phase.

2) In the optimization procedure, the following distinct phases can be emphasized:
   - The value imposed for the copolymer composition F_{1d} is fixed by the user.
   - The genetic algorithm computes the weights w attached to the objectives and the values of the decision variables t, T, C, f_1.
   - The neural model uses as inputs the four decision variables and provides predictions $-x$ and F_{1f}—necessary for evaluating the objective function.

3 Results and Discussion

Experimental data from Table 1 were used to train different neural networks, which have four inputs (t, T, C, f_1) and two outputs (x and F_1). 15% of these data represent a validation data set and the remaining data is the training data set.

Table 2 contains different feed forward topologies tested with selected training data and the main performances for these networks: *MSE* (Mean Squared Error), r (correlation between experimental and neural network outputs) and E_p (percent error). The structure of a network of MLP type (multilayer perceptron–feed forward neural network) is given by the number of neurons in the input layer, corresponding to the four input variables, then the number of hidden neurons

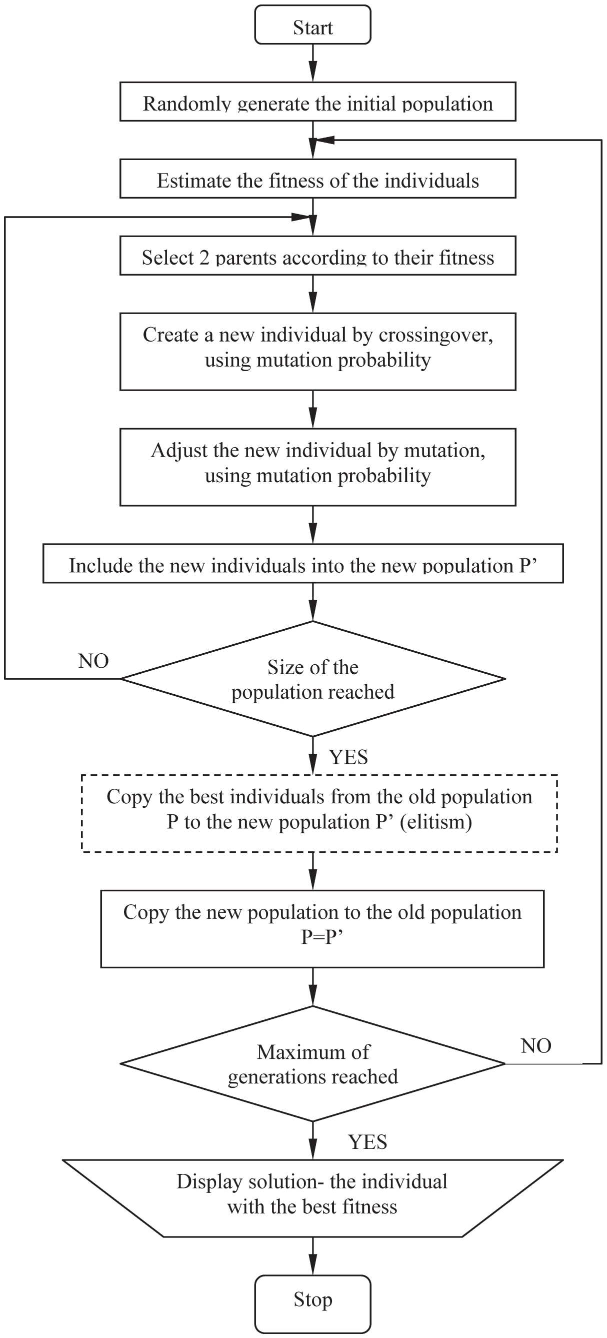

Start
Randomly generate the initial population
Estimate the fitness of the individuals
Select 2 parents according to their fitness
Create a new individual by crossingover, using mutation probability
Adjust the new individual by mutation, using mutation probability
Include the new individuals into the new population P’
Size of the population reached
NO
YES
Copy the best individuals from the old population P to the new population P’ (elitism)
Copy the new population to the old population P=P’
Maximum of generations reached
NO
YES
Display solution- the individual with the best fitness
Stop

Fig. 2. General flow chart of a genetic algorithm.

(in one or two layers) and, finally, the number of neurons in output layer for the output variables. The multi-layer perceptron (MLP) is the best known and most widely used kind of neural network. This paper also recommends MLP for polymerization reaction modeling.

Hidden neurons, as well as output layer neuron, use a hyperbolic tangent as nonlinear activation functions. All the network weights were initialized as random numbers in the interval (−0.5, 0.5). The networks were trained using the back-propagation algorithm. Once the data have been fed into the neural networks, the weights were updated continuously based upon the back propagation learning rule. The training phase was considered complete when the error of all the training patterns was less than a pre-specified error criterion or a maximum number of epochs (iterations) had been reached. If, after the entire set of training patterns was presented, the overall error was still unacceptable, the neural network would be returned to the beginning of the training patterns and the process would be repeated. So, the training is considered terminated at the point where network error (*MSE*) becomes sufficiently small.

Table 2. Different MLP topologies trained for the copolymerization process

No.	Topology	*MSE*	*r*	E_p
1	MLP (4:5:2)	0.000315	0.999374	2.1151
2	MLP (4:7:2)	0.000200	0.999529	0.7898
3	MLP (4:10:2)	0.000001	1	0.000017

The mean squared error was computed using the following formula:

$$MSE = \frac{1}{M \cdot L}\sum_{k=1}^{M}\sum_{p=1}^{L}\left(d_k^p - y_k^p\right)^2 \qquad (5)$$

where M is the number of nodes in output layer and L-number of exemplars in the data set (number of patterns), d_k^p is the desired output for exemplar p at processing element k and y_k^p is the network output for exemplar p at processing element k.

In this work, the number of hidden layers and units was established by training a different range of networks and selecting the one that best balanced generalization performance against network size. Consequently, a configuration of 4 input neurons, a single hidden layer with 10 neurons and an output layer with 2 neurons, noted MLP (4:10:2) was used.

The predictions of the neural network on the training data were compared to the experimental ones in order to verify how the model learned the behavior of the process.

Good predictions are obtained with the neural model MLP (4:10:2), on training data: average relative errors of 0.0027% for conversion and 0.00001% for copolymer composition. The correlation between experimental data and network predictions was more than 0.999. Relative errors were calculated using the following formula:

$$E_r = \frac{p_{exp} - p_{net}}{p_{exp}} \cdot 100 \qquad (6)$$

where p represents the parameter under study (conversion and copolymer composition), indexes *exp* and *net* denote experimental and neural network values.

Several examples are presented in Fig. 3 and 4, which show a comparison between two sets of data, experimental and network outputs.

Good agreement between the two data sets proves that the neural model has learned well the behavior of the process.

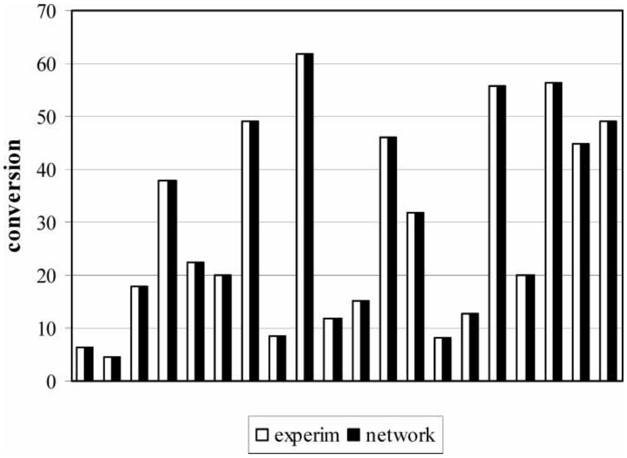


Fig. 3. Experimental data (white bars) and neural network results (black bars) obtained with MLP (4:10:2), in the training phase for reaction conversion.

The simple topology of the neural model (only one intermediate layer with 10 neurons), the small training time and the very good results obtained in training the model are related to the quality of the experimental data and the way in which they cover the experimental domain. A planned experiment is a guarantee that the reaction domain was covered evenly.

A key issue in neural network based process modeling is the robustness or generalization capability of the developed models, i.e., how well the model performs on unseen data. Thus, a serious examination of the accuracy of the neural network results requires the comparison with experimental data, which were not used in the training phase (previously unseen data). The predictions of the networks on validation data are given in Table 3.

One can notice a satisfactory agreement between the two categories of data, experimental and neural network predictions, with an average relative error less than 4%. For this reason, the projected neural model MLP (4:10:2) can be used to make predictions under different reaction conditions, and, also, can be included into the optimization procedure.

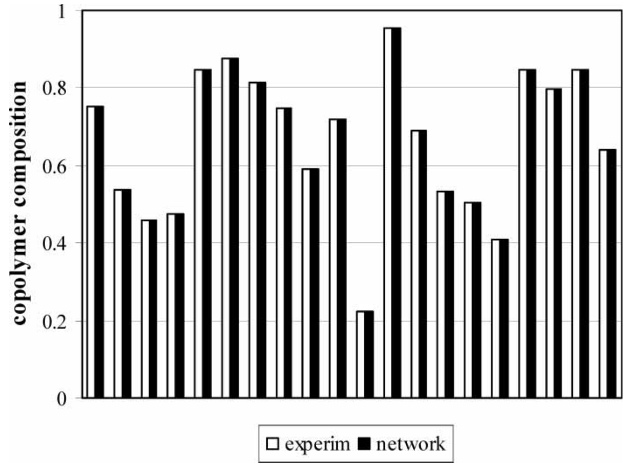


Fig. 4. Experimental data (white bars) and neural network results (black bars) obtained with MLP (4:10:2), in the training phase for copolymer composition.

Table 3. Results of the MLP (4:10:2) on validation data

				Experim	Network	Experim	Network
t	T	C	f_1	x	x	F_1	F_1
0.75	45	4.5	0.3	4.46	4.11	0.55	0.59
1.25	60	5.5	0.5	44.78	42.50	0.65	0.69
1.75	75	2.5	0.3	41.5	39.8	0.44	0.42
1.75	75	4.5	0.7	68.61	67.1	0.78	0.81
1.25	60	3.5	0.5	32.53	29.8	0.69	0.67

Process optimization is a very important in polymer industry being related to the production cost reduction and higher quality requirements. However, product quality is a much more complex issue in polymerization than in most conventional short chain reactions. Because the molecular architecture of the polymer is so sensitive to reactor operating conditions, such as upsets in reaction conditions can alter critical molecular properties. Because the main goal in operating a polymerization reactor is to produce a final polymer with certain chemical and mechanical properties, the requirement of an accurate process model for the optimal quality control is becoming very important. In addition, an optimization technique must be involved to obtain the reaction conditions that give certain values of an objective function.

A mathematical model can easily predict the polymer properties from the inputs of the reactor conditions. But the other way around (inverse modeling) is much more difficult to do and an optimization technique, highly time consuming, must be involved. The optimization approach can be substituted by a neural network trained to do the same job.

The inverse neural modeling that is the determination of reaction conditions that lead to pre-established properties, has as advantage the substitution of complex modeling and optimization procedure with a simple and rapid technique often supplying reliable results. Generally, once trained, the neural network can estimate the reactor operating conditions faster than the optimization algorithm.

Various neural network models are developed for predicting the reaction conditions for obtaining copolymers with prespecified compositions. The optimization problem based on inverse neural network modeling can be formulated in different ways, such as:

Problem 1. What is the initial composition of the reaction mixture and how much time is necessary to achieve a final desired composition for the copolymer, working at pre-established values of temperature and amount of catalyst?

Problem 2. What are the reaction temperature and time necessary to achieve a final desired composition for the copolymer, working at pre-established values of amount of catalyst and initial composition of the monomers?

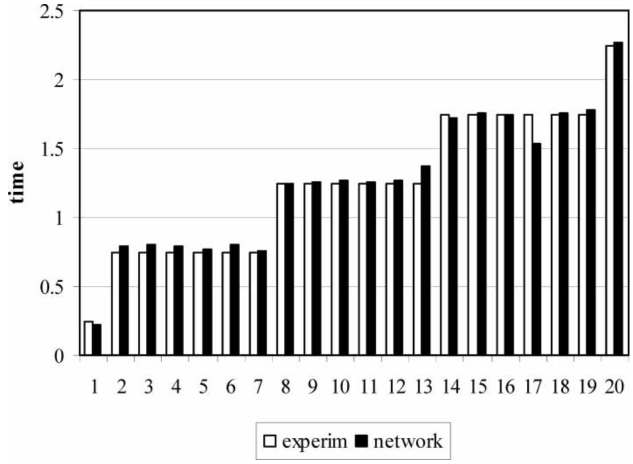


Fig. 5. Experimental data (white bars) and neural network results (black bars) obtained with MLP (3:10:2) in the training phase of inverse modeling for the reaction time.

Problem 3. What are the amount of catalyst and reaction time necessary to achieve a final desired composition for the copolymer, working at pre-established values of temperature and initial composition of the monomers?

One of the above problems, for instance problem 1, is exemplified in the following. In this case, the neural network which performs the inverse modeling has as inputs F_{1f}, T, C and as outputs f_1, t. In this optimization, the imposed parameters represent the objectives to be accomplished and the parameters computed by the neural model are decision variables.

The same methodology based on trial and errors is applied for determining the topology of the neural model. Different networks of MLP type are trained and, taking into account their performance, MLP (3:10:2), that means a network with one hidden layer with 10 neurons, is chosen. In the training phase, $MSE = 0.006123$, $r = 0.989$ and $E_p = 2.1685$. Figures 5 and 6 illustrate the comparison between experimental data and the results of the neural model obtained in the training phase.

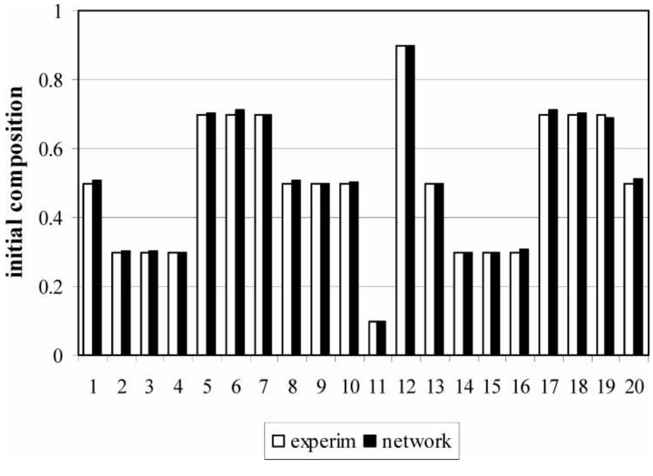


Fig. 6. Experimental data (white bars) and neural network results (black bars) obtained with MLP (3:10:2) in the training phase of inverse modeling for the initial composition of the reaction mixture.

Table 4. Validation of MLP (3:10:2) in inverse neural network modeling

T	C	F_1	Experim t	Experim f_1	Network t	Network f_1	Time error	f_1 error
45	4.5	0.55	0.75	0.3	0.79	0.29	5.33	3.33
60	5.5	0.654	1.25	0.5	1.29	0.48	3.2	4.00
75	2.5	0.439	1.75	0.3	1.69	0.32	3.42	6.66
75	4.5	0.777	1.75	0.7	1.71	0.68	2.28	2.85

Table 4 shows the validation phase for the inverse model MLP (3:10:2). Table 5 contains some recommendations obtained as predictions of inverse neural model for performing several experiments, according to the optimization problem 1: the final composition of the copolymer, temperature and amount of catalyst are established and one determines, by inverse neural modeling, the reaction time and the initial composition of the monomer mixture, which lead to the desired copolymer composition.

The key advantage of using a GA to search for the global optimum is the ability of the GA to do a multi-pronged population based search. The performance of a GA critically depends on the representation of the solutions, the definition of the genetic operators that transforms the solutions from one generation to another and the values of the algorithmic parameters such as the percentage of elitism and the probabilities of mutation and crossover. In this study, we use a floating point representation of the parameters. Rank-based selection procedure is used in every generation to identify the individuals to be manipulated by the GA operators.

Population size, number of generations, crossover probability and mutation probability are known as the control parameters of genetic algorithms. The values of these parameters must be specified before the execution of GA and they depend on the nature of the objective function.

The results of the optimization are represented by the values of the decision variables Equation (2) that lead to a minimum value of the objective function (Equation (3)), which means, from the point of view of partial objectives,

Table 5. Predictions of MLP (3:10:2) in inverse neural network modeling

T	C	F_1	t	f_1
45	2.5	0.2	2.36	0.1
45	2.5	0.4	2.36	0.2
45	2.5	0.8	2.36	0.9
50	2.5	0.2	2.36	0.1
50	2.5	0.5	2.36	0.4
50	2.5	0.8	1.68	0.4
40	3	0.4	2.36	0.2
40	3	0.6	0.14	0.3
40	3	0.8	2.36	0.9

the maximization of the conversion and the achievement of the imposed value for the copolymer composition.

The optimization procedure is implemented in *Matlab 7.5* with original software, as specific functions were programmed for each phase of the genetic algorithm.

One obvious problem of the optimization that combines many objectives into a single function is that it may be difficult to generate a set of weights (namely *w* in Equation (3)) that properly scales the objectives when little is known about the problem. A single weighted sum approach requires a priori knowledge of the weights to vary the emphasis given to each objective. The present approach has the advantage of computing the optimal values for the weights coefficients of the objectives within the genetic algorithm, along with the optimal values for decision variables.

The optimization results are presented in a series of tables with the following structure: column 1 contains the identification number used to refer the optimization in the discussions; columns 2, 3 and 4 contain the parameters of the genetic algorithm: the size of the initial population, pop_size, the number of generations, gen_no and the values for mutation and crossover rates, mut_rate, cross_rate; column 5—the weights of the objectives computed within the genetic algorithm; column 6—the optimal values of the decision variables provided by the GA; column 7—monomer conversion and copolymer composition obtained as predictions of the neural model; column 8—the value of the objective function and the imposed value for copolymer composition.

Since GA is a stochastic algorithm, we run it many times, for each situation (each row in the tables of results) to get statistically meaningful values for the time taken. Longer runs are not necessary as GA is already a population-based procedure that works with multiple solutions. One of the solutions (the best solution) is chosen and inserted into the tables.

Table 6 contains optimizations where different values for dimension of initial population (pop_size) and number of generations (gen_no) are chosen. GA research showed that the solution improves as the number of individuals in the

Table 6. Optimizations with different values for the two GA parameters: size of initial population and number of generations

No.	pop_size	gen_no	cross_rate mut_rate	Weights calculated by GA	Control variables	Outputs NN	J, F_{1d}
1	50	100	cross_rate = 0.9 mut_rate = 0.03	$w_x = 0.4725$ $w_{F1} = 7.4476$	$t = 2.17$ $T = 86.22$ $C = 3.90$ $F_1 = 0.55$	$x = 63.62$ $F_1 = 0.69$	$J = 0.172217$ $F_{1d} = 0.7$
2	50	500	cross_rate = 0.9 mut_rate = 0.03	$w_x = 0.1148$ $w_{F1} = 1.3562$	$t = 2.25$ $T = 96.33$ $C = 3.61$ $F_1 = 0.6$	$x = 64.57$ $F_1 = 0.69$	$J = 0.04103$ $F_{1d} = 0.7$
3	50	1000	cross_rate = 0.9 mut_rate = 0.03	$w_x = 0.0463$ $w_{F1} = 1.2506$	$t = 2.24$ $T = 96.57$ $C = 3.85$ $F_1 = 0.59$	$x = 64.26$ $F_1 = 0.7$	$J = 0.016534$ $F_{1d} = 0.7$
4	50	2000	cross_rate = 0.9 mut_rate = 0.03	$w_x = 0.0139$ $w_{F1} = 0.1613$	$t = 2.08$ $T = 109.91$ $C = 2.77$ $F_1 = 0.64$	$x = 64.59$ $F_1 = 0.7$	$J = 0.004914$ $F_{1d} = 0.7$
5	100	100	cross_rate = 0.9 mut_rate = 0.03	$w_x = 0.8618$ $w_{F1} = 42.6603$	$t = 1.9806$ $T = 89.52$ $C = 3.45$ $F_1 = 0.58$	$x = 63.65$ $F_1 = 0.7$	$J = 0.313386$ $F_{1d} = 0.7$
6	200	200	cross_rate = 0.9 mut_rate = 0.03	$w_x = 0.0802$ $w_{F1} = 11.2916$	$t = 1.99$ $T = 85.85$ $C = 3.39$ $F_1 = 0.58$	$x = 63.08$ $F_1 = 0.7$	$J = 0.029593$ $F_{1d} = 0.7$
7	500	200	cross_rate = 0.9 mut_rate = 0.03	$w_x = 0.0105$ $w_{F1} = 45.9843$	$t = 1.82$ $T = 94.57$ $C = 2.72$ $F_1 = 0.60$	$x = 63.84$ $F_1 = 0.7$	$J = 0.003799$ $F_{1d} = 0.7$
8	200	500	cross_rate = 0.9 mut_rate = 0.03	$w_x = 0.0048$ $w_{F1} = 5.2066$	$t = 2.19$ $T = 90.67$ $C = 3.18$ $F_1 = 0.6$	$x = 63.89$ $F_1 = 0.7$	$J = 0.001730$ $F_{1d} = 0.7$

population increases, but only up to a point. Beyond that, a larger population decreases the convergence speed of the algorithm, without leading to an improvement of the solution. With the increase in the number of generations, the execution time increases. Since the GA is an iterative procedure, the quality of the solution should increase with the number of generations, especially if elitism is used, which guarantees the fact that the solution will not worsen over time. But for each parameter and process there is a limit beyond which there are no more improvements of the results.

Some observations have to be made for the results presented in Table 6, without the choice of GA parameters.

In all cases, 1–8, $F_1 = F_{1d}$, that is one of the optimization goals, but the reaction conversion is relatively small. In the case of the dimethylmethylvinylsiloxane copolymers, the relatively low values for conversion can be explained by reversibility of the process. The position of the equilibrium depends on some factors between them being also the silicon substituent nature. Thus, by increasing the bulk of the substituents, the equilibrium shifts to the left and, as a result, the conversion will be lower. The experimental data showed in Table 1 reflect such a situation: the maximum conversion is 68% irrespective of the used parameter values combination.

The solution of a multi-objective optimization can be considered from two points of view: mathematically (minimum value for the objective criterion, J) and technologically, by appreciating each individual objective. In Table 6, the parameters pop_size and gen_no were chosen accordingly with

Table 7. Optimizations with different values for the two GA parameters: crossover rate and mutation rate

No.	pop_size	gen_no	cross_rate mut_rate	Weights calculated by GA	Control variables	Outputs NN	J, F_{1d}
1	50	500	cross_rate = 0.9 mut_rate = 0.03	$w_x = 0.1148$ $w_{F1} = 1.3562$	$t = 2.25$ $T = 96.33$ $C = 3.61$ $F_1 = 0.6$	$x = 64.27$ $F_1 = 0.69$	$J = 0.041033$ $F_{1d} = 0.7$
2	50	500	cross_rate = 0.8 mut_rate = 0.03	$w_x = 0.4460$ $w_{F1} = 0.7236$	$t = 2.18$ $T = 102$ $C = 3.62$ $F_1 = 0.61$	$x = 64.41$ $F_1 = 0.7$	$J = 0.158722$ $F_{1d} = 0.7$
3	50	500	cross_rate = 0.5 mut_rate = 0.03	$w_x = 0.0193$ $w_{F1} = 0.4136$	$t = 2.23$ $T = 100.66$ $C = 3.98$ $F_1 = 0.59$	$x = 64.35$ $F_1 = 0.7$	$J = 0.006878$ $F_{1d} = 0.7$
4	50	500	cross_rate = 0.1 mut_rate = 0.03	$w_x = 0.0293$ $w_{F1} = 0.1532$	$t = 2.23$ $T = 91.34$ $C = 4.35$ $F_1 = 0.55$	$x = 64.01$ $F_1 = 0.7$	$J = 0.010551$ $F_{1d} = 0.7$
5	50	500	cross_rate = 0.9 mut_rate = 0.1	$w_x = 0.0980$ $w_{F1} = 0.2540$	$t = 2.17$ $T = 101.74$ $C = 2.53$ $F_1 = 0.61$	$x = 64.40$ $F_1 = 0.7$	$J = 0.034874$ $F_{1d} = 0.7$
6	50	500	cross_rate = 0.9 mut_rate = 0.5	$w_x = 0.2500$ $w_{F1} = 41.4896$	$t = 1.78$ $T = 107.81$ $C = 3.82$ $F_1 = 0.61$	$x = 64.42$ $F_1 = 0.7$	$J = 0.08874$ $F_{1d} = 0.7$
7	50	500	cross_rate = 0.9 mut_rate = 0.8	$w_x = 0.0539$ $w_{F1} = 70.9907$	$t = 1.09$ $T = 79.71$ $C = 3.11$ $F_1 = 0.56$	$x = 55.05$ $F_1 = 0.7$	$J = 0.024970$ $F_{1d} = 0.7$
8	50	500	cross_rate = 0.9 mut_rate = 0.01	$w_x = 0.2624$ $w_{F1} = 0.1705$	$t = 2.10$ $T = 109.72$ $C = 2.59$ $F_1 = 0.63$	$x = 64.59$ $F_1 = 0.7$	$J = 0.092923$ $F_{1d} = 0.7$
9	50	500	cross_rate = 0.9 mut_rate = 0.05	$w_x = 2.4342$ $w_{F1} = 2.9994$	$t = 2.18$ $T = 108.33$ $C = 2.85$ $F_1 = 0.63$	$x = 64.57$ $F_1 = 0.7$	$J = 0.862324$ $F_{1d} = 0.7$

the conversion value. Even the higher values for GA parameters lead to smaller values for J, the conversion did not become better. Consequently, pop_size = 50 and gen_no = 500 are sufficient for our goal.

The crossover rate (cross_rate) represents the probability with which a new individual is generated from two parents. If the rate is small, there is a high chance that one of the parents will be directly copied into the new population. Since crossover is the basis of the search process, a rate close to 1 should increase the speed of finding a solution. As a rule of thumb, the crossover probability is generally greater than 0.75 so as to encourage better exploration of the search space. Copying a parent into the new population is beneficial only when it has a high fitness value (the elitism achieves this objective in order not to lose the best solutions). Table 7 motivates the choice of the 0.9 rate, based both on partial objectives, and on the minimum value of the objective function.

Particularly, for the process under study, the GA parameters do not have significant influence on the optimization results. In Table 7, one can see that different values for the crossover and mutation rates lead to similar results for reaction conversion and copolymer composition. Only a high mutation rate in optimization no. 7 worsens the conversion value.

Therefore, the appropriate parameters of GA used to solve the proposed optimization problem are: pop_size = 50, gen_no = 500, cross_rate = 0.9 and mut_rate = 0.03.

The use of a scalar objective function that weightily combined its partial objectives presents disadvantages, as well as advantages. The main disadvantage comes from the weight coefficients, which have to be previously known. As advantages, one can mentioned the simplicity of the optimization solving based on a scalar function and that the user can give priority to one of the partial objective by manipulating the values of the weights. In this work, the disadvantage of the scalarization is counteracted by the fact that GA computes the weight coefficients along with the optimal values of the control variables. This is possible because the GA considers the optimization problem globally, irrespective of the particular meaning given to the chromosomes.

Table 8 contains some optimizations where the user imposes certain weights, depending on the objective with greater importance and based on accumulated knowledge about the process.

Even if the user imposes a high value for conversion weight, the conversion value does not exceed 64%. The synthesis process of dimethylmethylvinylsiloxane copolymers is not sensitive to weight values in the scalar optimization procedure.

The last table (Table 9) contains optimizations performed with different weights, imposed by the user or computed within the GA method, and with different values of F_{1d}. When GA provides optimal values for the weight coefficients, one of the goals of the optimization is achieved: $F_{1f} = F_{1d}$.

The optimization procedure based on a simple genetic algorithm and a neural network model applied in this paper is easy to manipulate and provides satisfactory results. In this way, a theoretical analysis of the synthesis of statistical dimethylmethylvinylsiloxane copolymers approached here is performed, with useful information for the practical application.

Table 8. Optimizations performed with the imposed values for the weight coefficients

No.	pop_size	gen_no	cross_rate mut_rate	Weights coefficients	Control variables	Outputs NN	J, F_{1d}
1	50	500	cross_rate = 0.9 mut_rate = 0.03	$w_x = 0.5$ $w_{F1} = 2$	$t = 2.10$ $T = 109.75$ $C = 2.61$ $F_1 = 0.63$	$x = 64.59$ $F_1 = 0.7$	$J = 0.177052$ $F_{1d} = 0.7$
2	50	500	cross_rate = 0.9 mut_rate = 0.03	$w_x = 5$ $w_{F1} = 1$	$t = 2.08$ $T = 109.86$ $C = 2.78$ $F_1 = 0.64$	$x = 64.59$ $F_1 = 0.7$	$J = 1.77015$ $F_{1d} = 0.7$
3	50	500	cross_rate = 0.9 mut_rate = 0.03	$w_x = 10$ $w_{F1} = 0.5$	$t = 2.09$ $T = 109.65$ $C = 2.79$ $F_1 = 0.64$	$x = 64.59$ $F_1 = 0.7$	$J = 3.540571$ $F_{1d} = 0.7$
4	50	500	cross_rate = 0.9 mut_rate = 0.03	$w_x = 50$ $w_{F1} = 0.1$	$t = 2.18$ $T = 109.91$ $C = 2.83$ $F_1 = 0.69$	$x = 64.60$ $F_1 = 0.75$	$J = 17.70017$ $F_{1d} = 0.7$
5	50	500	cross_rate = 0.9 mut_rate = 0.03	$w_x = 100$ $w_{F1} = 0.1$	$t = 2.24$ $T = 109.95$ $C = 2.86$ $F_1 = 0.72$	$x = 64.60$ $F_1 = 0.78$	$J = 35.39882$ $F_{1d} = 0.7$

Table 9. Optimizations performed with different values pre-established for the final composition of the copolymers

No.	pop_size	gen_no	cross_rate mut_rate	Weights coefficients	Control variables	Outputs NN	J, F_{1d}
1	50	500	cross_rate = 0.9 mut_rate = 0.03	$w_x = 0.2692$ $w_{F1} = 1.3656$ GA weights	$t = 2.23$ $T = 108.27$ $C = 2.90$ $F_1 = 0.34$	$x = 64.38$ $F_1 = 0.3$	$J = 0.095898$ $F_{1d} = 0.3$
2	50	500	cross_rate = 0.9 mut_rate = 0.03	$w_x = 50$ $w_{F1} = 0.1$ user weights	$t = 2.24$ $T = 109.9$ $C = 2.69$ $F_1 = 0.43$	$x = 64.55$ $F_1 = 0.4$	$J = 17.73202$ $F_{1d} = 0.3$
3	50	500	cross_rate = 0.9 mut_rate = 0.03	$w_x = 0.5302$ $w_{F1} = 0.0973$ GA weights	$t = 0.47$ $T = 108.73$ $C = 3.05$ $F_1 = 0.0044$	$x = 57.88$ $F_1 = 0.1$	$J = 0.329219$ $F_{1d} = 0.1$
4	50	500	cross_rate = 0.9 mut_rate = 0.03	$w_x = 0.2035$ $w_{F1} = 0.4898$ GA weights	$t = 2.23$ $T = 85.18$ $C = 4.18$ $F_1 = 0.4$	$x = 63.24$ $F_1 = 0.5$	$J = 0.07483$ $F_{1d} = 0.5$
5	50	500	cross_rate = 0.9 mut_rate = 0.03	$w_x = 50$ $w_{F1} = 0.1$ user weights	$t = 2.11$ $T = 109.99$ $C = 2.70$ $F_1 = 0.52$	$x = 64.58$ $F_1 = 0.55$	$J = 17.70738$ $F_{1d} = 0.5$
6	50	500	cross_rate = 0.9 mut_rate = 0.03	$w_x = 0.0874$ $w_{F1} = 0.1875$ GA weights	$t = 2.22$ $T = 109.84$ $C = 3.28$ $F_1 = 0.87$	$x = 64.56$ $F_1 = 0.9$	$J = 0.030962$ $F_{1d} = 0.9$
7	50	500	cross_rate = 0.9 mut_rate = 0.03	$w_x = 50$ $w_{F1} = 0.1$ user weights	$t = 2.23$ $T = 109.56$ $C = 2.91$ $F_1 = 0.74$	$x = 64.59$ $F_1 = 0.80$	$J = 17.70267$ $F_{1d} = 0.9$

The modeling and optimization procedures based on NN and GA can be also applied to other processes for which the amount of knowledge is limited.

4 Conclusions

This paper provides a general and simple optimization strategy, based on genetic algorithms and neural networks, applied to a complex polymerization process. The GA solves the optimization problem and computes the weights attached to the partial objectives, and NN constitutes the model included in the optimization procedure.

Simple architecture neural networks (feedforward networks with one hidden layer) and simple methods of establishing the networks' structure are used in this paper. A series of experiments planned by a composed centered second order program is carried out to obtain statistical dimethyl-methylvinylsiloxane copolymers and represents the training and validation data sets for the networks. Accurate results obtained as predictions of neural model on previously unseen data (validation data) prove an acceptable generalization capability of the model and make it reliable for the optimization.

Regarding the optimization procedure, several issues can be underlined. The association between a scalar objective function with the objectives weightily combined and a GA solving method represents a recommended optimization method because of its simplicity and accurate results. Additionally, the procedure adopted here has the advantage of determining the objective weights within the GA algorithm, along with the optimal control variables, and is useful in cases where little knowledge is available about the system.

The inverse neural network modeling represents a particular variant of optimization, useful for the synthesis of polymers with pre-ordained properties. In our approach, one estimates the reaction conditions that lead to a copolymer with desired composition.

The article especially addresses the establishment of a general strategy for solving a multiobjective optimization problem using genetic algorithms. This strategy is quite general and could be applied to other polymerization processes, with a high probability to obtain accurate results.

5 References

1. Shahaf, G. and Marom, S. (2001) *J. Neuroscience*, **21(22)**, 8782–8788.
2. Zhang, J., Morris, A.J. and Martin, E.B. (1998) *Comput. Chem. Eng.*, **22**, 1051–1063.
3. Kuroda, C. and Kim, J. (2002) *Neurocomputing*, **43**, 77–89.
4. Curteanu, S., Leon, F. and Gâlea, D. (2003) *Eur. Chem. Technol. J.*, **5**, 225–231.
5. Curteanu, S. (2004) *Cent. Eur. J. Chem.*, **2**, 113–140.
6. Araújo, P.H.H., Sayer, C., De la Cal, J.C., Asua, J.M., Lima, E.L. and Pinto, J.C., II. ENPROMER: Florianópolis, Brazil, 1999.
7. Zhang, J., Martin, E.B., Morris, A.J. and Kiparissides, C. (1997) *Comput. Chem. Eng.*, **21**, s1025–s1030.
8. Zhang, J. (1999) *Neurocomputing*, **25**, 93–113.
9. Fernandes, F.A.N. and Lona, L.M.F. (2002) *Polym. Reaction Eng.*, **10**, 181–192.
10. Zhang, J., Morris, A.J., Martin, E.B. and Kiparissides, C. (1998) *Chem. Eng. J.*, **69**, 135–143.
11. Tian, Y., Zhang, J. and Morris, J. (2001) *Ind. Eng. Chem. Res.*, **40**, 4525–4535.
12. Tian, Y., Zhang, J. and Morris, J. (2002) *Chem. Eng. Proc.*, **41**, 531–538.
13. Zhang, J. (2004) *Ind. Eng. Chem. Res.*, **43**, 1030–1038.
14. Nascimento, C.A.O., Giudici, R. and Guardani, R. (2000) *Comp. Chem. Eng.*, **24**, 2303–2314.
15. Wei, J., Xu, Y. and Zhang, J. (2002) *Proceedings of the 2002 IEEE International Conference on Control Applications*, Glasgow Scotland, U.K.
16. Ng, C.W. and Hussain, M.A. (2004) *Chem. Eng. Proc.*, **43**, 559–570.
17. Horn, J. (2001) *Comput. Chem. Eng.*, **25**, 1561–1567.
18. Curteanu, S. and Popa, A.M. (2004) *Rev. Roum. Chim.*, **49**, 3–23.
19. Fernandes, F.A.N. and Lona, L.M.F. (2005) *Braz. J. Chem. Eng.*, **22**, 323–330.
20. Garg, S., Gupta, S.K. and Saraf, D.N. (1999) *J. Appl. Polym. Sci.*, **71**, 2101–2120.
21. Ahn, S.M., Chang, S. and Rhee, H.K. (1998) *J. Appl. Polym. Sci.*, **69**, 59–72.
22. Mitra, K., Majumdar, S. and Raha, S. (2004) *Comp. Chem. Eng.*, **28**, 2583–2594.
23. Silva, C.M. and Biscaia, E.C., Jr. (2003) *Comp. Chem. Eng.*, **27**, 1329–1344.
24. Zhou, F., Gupta, S.K. and Ray, A.K. (2000) *J. Appl. Polym. Sci.*, **78**, 1439–1458.
25. Coello, C.A. *Congress on Evolutionary Computation*; IEEE Service Center: Piscataway, New Jersey, Vol. 1. 30–37, 2000.
26. Deb, K. *Multi-objective Optimization Using Evolutionary Algorithms*; Wiley: Chichester, U.K., 2001.
27. Carlos, A.C.C., van Veldhuizen, D.A. and Lamont, G.B. *Evolutionary Algorithms for Solving Multi-objective Problems*; Kluwer Academic: New York, 2002.
28. Bhaskar, V., Gupta, S.K. and Ray, A.K. (2000) *Rev. Chem. Eng.*, **16**, 1–54.
29. Nandasana, A.D., Ray, A.K. and Gupta, S.K. (2003) *Int. J. Chem. React. Eng.*, **1**, 1–16.
30. Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T.A. (2002) *IEEE Trans. Evol. Comut.*, **6**, 182–195.
31. Bauer, J., Nicola, H. and Kickelbick, G. (2001) *Chem. Commun.*137–138.
32. Cazacu, M., Marcu, M., Holerca, M., Petrovan, S. and Lazarescu, S. (1996) *J. Macromol. Sci. Pure Appl. Chem.*, **A33**, 65–76.
33. Cazacu, M. and Marcu, M. (1995) *Macromol. Rep.*, **A32**, 1019–1029.
34. Cazacu, M., Marcu, M., Petrovan, S., Holerca, M. and Simionescu, M. (1996) *Polym. Plast. Technol. Eng. J.*, **35**, 327–347.
35. Vega, M.P., Lima, E.L. and Pinto, J.C. (2000) *Brazilian J. Chem. Eng.*, **17**, 471–481.
36. Won-Hyouk, J., Juergen, H. and Kenneth, H. (2005) *Comp. Chem. Eng.*, **30**, 285–294.
37. Torabi, S.A., Fatemi Ghomi, S.M.T. and Karimi, B. (2006) *Eur. J.Oper. Res.*, **173**, 173–189.