Revue Roumaine de Chimie, **2008**, *53*(3), 247–255

MULTIOBJECTIVE OPTIMIZATION OF A FREE RADICAL POLYMERIZATION. III. A HYBRID METHOD BASED ON GENETIC ALGORITHMS

Silvia CURTEANU*

"Gh. Asachi" Techical University, Faculty of Industrial Chemistry
Bd. D. Mangeron No. 71A, 700050 IAŞI, ROUMANIA

Received March 15, 2007

This paper continues an optimization study[1,2] developing a hybrid method which combines genetic and gradient-based algorithms. This procedure was used with a scalar multiobjective function in which the weight coefficients attached to partial objectives were appreciated by successive tests or within the genetic algorithm method. A free radical polymerization process was chosen as case study, having as main objectives the maximization of the monomer conversion, the minimization of the reaction time and polydispersity index and the achievement of an imposed value for the number average polymerization degree. The decision variables of the optimization problem were the initial initiator concentration and the temperature of the process composed from one or several isothermal steps.

INTRODUCTION

In complex chemical processes, the ability to select optimum operating conditions in the presence of multiple conflicting objectives, given the various constraints of the process, can present a major challenge. The optimization of a free-radical polymerization is multiobjective in nature, since it normally has several objectives, often conflicting and non-commensurable, that must be satisfied at the same time.

Traditionally, optimization techniques have dealt with multiple objectives combining them into a single objective function composed of the weighted sum of the individual objectives, or by focusing on one objective while transforming the others into constraints.[3] The use of a scalar objective function[4,5] allows simple algorithms to be used for solving the problem, but suffers from several drawbacks. One is that the results depend largely on the values of the weighting factors used, which are difficult to assign on a priori basis. What is even more important is that there is a risk of loosing some optimal solutions.[6]

Computer-aided optimization methods have been widely employed in chemical process industries. The traditional optimization techniques, direct or gradient-based methods, have, as main disadvantage, the fact that the convergence to an optimal solution depends on the chosen initial solution and most algorithms tend to get stuck to a sub-optimal solution. In case of complex polymerization problems, the search space often have multiple optima and, if an initial guess is provided near to a local optima, traditional algorithms converge to one of those local optima.

In recent years, an extremely robust technique, genetic algorithm (GA) and its adaptations for more useful but complex multiobjective optimization problems, have become popular because of some important advantages such as they are not sensitive to starting point and they are more likely to find out a function's true global optimum.[7]

Hybridization is an approach that combines a gradient-based algorithm and a genetic algorithm to improve search performance.[8-13] One of the most popular examples is an algorithm that uses GA to search for solution across the entire space, whereas the gradient-based search determines local optima starting from a population member. GAs often perform well in the global search, but they are relatively slow in converging to a local optima.[14] On the other hand, the local improvement methods

* Corresponding author: scurteanu@ch.tuiasi.ro; silvia_curteanu@yahoo.com

can find the local optimum in a small region of the search space, but they are typically poor in a global search. Therefore, various strategies of hybridization have been suggested to improve performance of simple GAs.[14-16] These hybridizations usually involve incorporating a neighborhood search heuristic as a local improver into a basic GA loop of recombination and selection. That is, local improver is applied to each newly generated offspring to move it to a local optimum one before inserting it into the enlarged population. In this manner, GA is used to perform global exploration among a population (*i.e.* as a diversification tool), while the local improver is used to perform local exploitation around chromosomes (*i.e.* as a intensification tool).

The first part of our study entitled "Multiobjective optimization of a free radical polymerization" uses a multiobjective function in a scalar form and a traditional solving method based on Sequential Quadratic Programming (SQP).[1] The second part of the study develops an optimization method based on genetic algorithms applied to free radical polymerization of methyl methacrylate.[2]

In this paper (the third part), in order to search for the global optimum, a hybrid method (HM) have been proposed where a genetic algorithm is used to identify initial guesses for a local optimizer used to determine the optimum. The local optimizer for refining GA solutions is the SQP method. Therefore, our simple hybrid technique consists in a combination of GA and SQP, successively used. The difference between our hybridization technique and those reported in the literature consists in a way of combining the two methods, resulting a simple optimization procedure, suitable to work with scalar objective function. New elements in the optimization procedure are represented by the objective weights calculated within the genetic algorithm, along with optimal values for decision variables, which means an argument for the use of a scalar objective function. Polymerization processes, with a series of difficulties related to modeling and optimization methodologies, are appropriate examples for testing a new variant of optimization, considered to be efficient. Different types of GA, with different parameters, are used in optimization in order to chose the method suitable for the process under study. A comparison between the three solving methods, SQP, GA and HM is also provided.

OPTIMIZATION TECHNIQUES: GA, SQP AND HM

GAs are intelligent stochastic optimization techniques based on the mechanism of natural selection and genetics. They start with an initial set of solutions, called population. Each solution in the population is called a chromosome (or individual), which represents a point in the search space. The chromosomes are evolved through successive iterations, called generations, by genetic operators (selection, crossover and mutation) that mimic the principle of natural evolution. 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 offsprings, are created and survive with chromosome in the current population, called parents, to form a new population.

In a typical genetic algorithm, the standard representation of solutions is an array of bits. In our GA model, we used real values encoding for the chromosomes, which is more time efficient, with better precision of the solutions. The optimization tests are based on some variants of GA resulted from different methods for selection, crossover and mutation.

For selection phase, the following procedures were used: roulette wheel selection, rank selection and tournament selection. In the roulette wheel selection the parents are selected proportionally to their fitness. Rank selection first ranks the population and then every chromosome receives fitness from this ranking. Both the roulette wheel and the rank selections have the disadvantage of being computationally expensive. In tournament selection, some individuals are selected at random from the population and the fittest of them is selected. The most common type of tournament selection is binary tournament selection, where just two individuals are selected to produce a child.

Arithmetic real value crossover produces a linear combination of the parents. Given a uniform random number $r \in [0,1]$,

$$C = r \cdot M + (1 - r) \cdot F \quad \text{or} \quad C = r \cdot F + (1 - r) \cdot M \tag{1}$$

where *C* is the real value chromosome of the child, and *M* and *F* are the chromosomes of the parents. Two variants of crossover were used in this study: with the same point for all genes and with different

points for the genes. If we imagine the individuals as point in an *n*-dimensional space of the features produced by the genes, a crossover with the same point generates a new individual on the line segment that links the two parent points. This means that the *r* above is the same for all genes. Therefore, the offspring will be like its parents to the same extent for all its features. In the case of second variant, the new individual will no longer be on the line segment that links its parents. The *r* value is different for every gene. The offspring will look more like one parent regarding a feature and less regarding another.

Two variants of mutation - fine tuning and resetting – were included in the optimization procedure. In variant 1 of mutation - fine tuning - after a new individual has been created, a mutation is performed on it. Given the chosen solution encoding, a uniform mutation can be employed, that randomly changes a gene to a uniform random value from an interval $[0.95g_i, 1.05g_i]$, where g_i represents the current value of the gene. We did not use absolute boundaries for the interval in order not to constrain this genetic operator. In this way, the individual is "shaken" a little; after the mutation it receives a gene value close to the one obtained by crossover. Thus, catastrophic mutations are avoided, that totally change a gene value. This method performs a "fine tuning" of the gene value, and thus an individual close to the optimal solution will not be taken out of that area in the solution space.

Variant 2 – resetting - is conceptual opposite of the previous variant. A gene value is reset to a random value in its search interval. The purpose is to refresh the search process, in 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).

Within the hybrid approach, GA is followed by SQP which mimics Newton's method for constrained optimization in that at each major iteration an approximation is made of the Hessian of the Lagrangian function using a quasi-Newton updating method. This is then used to generate a QP sub-problem whose solution forms a search direction for a line search procedure.[1]

In HM, the solution of GA represents the starting point for SQP. HM combines the advantages of both the above mentioned methods. The main drawbacks of SQP – local optima and initial guesses required – are avoided by the GA. It is widely acknowledged that the performances of a GA can be improved by combining problem-specific knowledge for particular problems, especially for large-scale hard optimization problems. Thus, not only the exploration ability of a GA can be enhanced by using multiple searching operators so as to explore the solution space in a better global sense, but also the exploitation ability can be enhanced by using local search based on problem information so as to concentrate the computing effort on more promising neighbor solutions.

The association between a scalar objective function and a hybrid solving method for optimization problem removes the drawbacks of the "scalarization": 1) GA avoids the local minima; 2) when necessary, the optimal values for the weights of the individual objectives can be calculated within the genetic algorithm, along with the optimal values for the decision variables. This study uses two methods for calculating the weights attached to partial objectives – trial and errors method and within the GA optimization.

In that follows, the efficiency of the optimization technique based on a scalar objective function and on a hybrid solving method that combines the advantages of SQP and GA is demonstrated.

RESULTS AND DISCUSSION

A good process model is a necessary prerequisite for application of the optimal control strategy. Consequently, the kinetic model has been validated by experimental runs of the bulk polymerization in a wide range of operating conditions. Our previous works [17,18] present good agreement between simulation results and experimental data for the kinetic model developed for methyl methacrylate (MMA) polymerization. This process was chosen to illustrate the HM optimization because it is well known and a good kinetic model is available.

At each step of the optimization procedure, model equations are integrated using a special function for solving stiff differential equation, ode15 in Matlab 7.0. Integration leads to conversion, number and weight average molecular weight histories for the time interval of the reaction.

The performance index *J* is defined by the following equations:

$$J = w_t \cdot t_f + w_Q \cdot Q_f + w_x \cdot (1 - x_f) + w_{DPn} \cdot \left(1 - \frac{DP_{nf}}{DP_{nd}}\right)^2 \quad (2)$$

where w are weighting factors, Q is the polydispersity index ($Q = DP_w/DP_n$), DP_{nd} is the desired value of number average polymerization degree at $t = t_f$; x_f and DP_{nf} are the actual values corresponding to the final reaction time t_f. DP_w is the weight average polymerization degree. Minimization of the objective function, J, means minimum reaction time, maximum monomer conversion, minimum polydispersity index and obtaining a polymerization degree DP_{nf} close to the desired value, DP_{nd}.

The decision variables are the initial concentration of the initiator, I_0, and the reaction temperature represented by three isothermal steps, T_1, T_2, T_3. The polymerization process is conducted in a fixed time of 400 s, in a perfect mix batch reactor. Limit ranges for the reaction temperature, T, and the initiator concentration feed, I_0, are established based on experimental data of MMA bulk free radical polymerization: $40°C \le T \le 90°C$ and $10 \text{ mol/m}^3 \le I_0 \le 50 \text{ mol/m}^3$.

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

A major drawback of using traditional GA as a search procedure is that there is no rigorous mathematical proof that it will converge to the global solution or can obtain all the local optima. However, we only use the GA to identify the promising regions of parameter space that are later explored by local optimizer (SQP).

We can take into account that the performances of a GA often depend on its parameters and operators, and it is easy to cause premature convergence. To achieve superior optimization performances, the genetic operators need to be well designed especially so as to enhance and balance the exploration and exploitation abilities. In order to detect the best set of GA parameters, along with a parallel comparison of the three optimization methods (SQP, GA and HM), an objective function with a small number of restrictions is initially chosen:

$$J = w_Q \cdot Q_f + w_x \cdot (1 - x_f) \quad (3)$$

with the weights w established based on our previous tests.[1] Minimization of index J represents minimization of the polydispersity index and maximization of the monomer conversion.

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 tables that show the results of the optimizations contain the following columns: the current number and the method used, the values of decision variables, I_0 and T (three values for temperature, T_1, T_2, T_3), conversion, x, time, t, polydispersity index, Q, number average polymerization degree, DP_n (all these resulting from solving the model in the conditions I_0, T established through the optimization procedure), the value of the objective function and an observation column where the parameters of the methods are mentioned. In an optimization, three values are shown for conversion and time: the first two represent the values of the intermediate steps of temperature, and the last – the value obtained at the end of the reaction, when using optimal parameters.

Table 1 emphasizes the influence of the population dimension (*dim*) on the optimization results, using different methods for solving the optimization problem (GA, SQP and HM). The selection method is the rank selection, and for the crossover and mutation, the variants marked as 1 in the section that describes the GA are used. Other values being used are: the crossover rate (*cross*) = 0.8, the mutation rate (*mut*) = 0.03, and the number of generations (*gen*) = 15. The weights attached to the partial objectives are $w_Q = 3$ and $w_x = 10$. Also, starting point for SQP is written; for instance, [90, 90, 90, 0.1, 0.7, 10] represent: $T_1 = 90°C$, $T_2 = 90°C$, $T_3 = 90°C$, $x_1 = 0.1$, $x_2 = 0.7$, $I_0 = 10 \text{ mol/m}^3$. The two runs SQP (1SQP and 2SQP) differ trough the starting point. Their results can be compared with the GA and HM results.

From the point of view of solving method, HM provide the best results. Comparing 1GA with 1SQP and 1HM, minimum J corresponds to 1HM and, also, good values for conversion (0.92) and polydispersity index (2.43). 2HM represents a significant improvement compared to 2GA because of increasing of conversion from 0.85 to 0.91 and decreasing of polydispersity from 2.81 to 2.42; objective function in 2HM is 8.126 and in 2GA, 9.973.

Table 1

The influence of the initial population dimension upon optimization results

No.	I_0 (mol/m^3)	T (^{o}C)	x	t (min)	Q	DP_n	Objective function	Observations
1 GA	15.6	61.4 85 36.5	0.46 0.85 0.90	117 121 400	2.41	3961	8.2266	**dim=20**
1 SQP	10	90 90 90	0.1 0.7 0.98	5 27 60	4.78	1061	14.4971	[90, 90, 90, 0.1, 0.7, 10]
1 HM	15.6	61.4 85 40	0.44 0.85 0.92	116 121 500	2.43	3677	8.1200	[61.4, 85, 36.5, 0.46, 0.85, 15.6]
2 GA	18.8	45.9 63 62.3	0.40 0.80 0.85	382 395 400	2.81	1039	9.9733	**dim=50**
2 SQP	10	79.9 59.2 90	0.1 0.1 0.98	10 10 64	4.72	1086	14.3206	[90, 40, 90, 0.3, 0.5, 10]
2 HM	10	69.2 90 58.6	0.45 0.85 0.91	79 83 129	2.42	3239	8.1261	[45.9, 63, 62.3, 0.4, 0.8, 18.8]

GA research showed that the solution improves as the number of individuals in the population increases, but only up to a point. Beyond that, a larger population decreases the convergence speed, without leading to an improvement of the solution. In table 1, value dim = 20 corresponds to a minimum for objective function. For establishing the *dim* value, much more optimizations were performed; table 1 illustrates only a few examples. This statement is available for all the tables presented in this paper.

In table 2, different values for the number of generations are tested. With the increase in the number of generations (*gen*), 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. As table 2 shows, for MMA polymerization, a number of 50 generations (gen=50) is sufficient to provide acceptable results.

The crossover rate (*cross*) represents the probability with which from two parents a new individual is generated. If the rate is small, there are high chances that one of the parents to 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. 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 3 motivates the choice of the 0.8 rate, based both on partial objectives, and the minimum value of the objective function. In all three cases, HM finishes with smaller values for the objective function than GA and acceptable values for partial objectives.

Table 2

The influence of the number of generations upon optimization results

No.	I_0 (mol/m^3)	T (^{o}C)	x	t (min)	Q	DP_n	Objective function	Observations
1 GA	26	93.6 111 74.2	0.33 0.92 0.96	8 13 44	2.62	375	8.3063	dim=20 **gen=50**
1 HM	26.3	86.6 90.0 71.2	0.41 0.91 0.95	16 24 59	3.48	792	8.2947	[93.6, 111, 74.2, 0.41, 0.91, 26.3]
2 GA	20.5	42.6 73.2 81.2	0.23 0.75 0.85	376 399 400	3.36	3984	9.5804	dim=20 **gen=100**
2 HM	12.4	51.3 73.6 69.3	0.2 0.21 0.95	178 179 268	4.89	3862	9.5354	[42.6, 73.2, 81.2, 0.23, 0.75, 20.5]

Table 3

The influence of the crossover rate upon optimization results

No.	I_0 (mol/m^3)	T (^{o}C)	x	t (min)	Q	DP_n	Objective function	Observations
1 GA	19.8	71.9 94.7 59	0.50 0.83 0.92	53 55 107	2.45	1826	8.2152	dim=20 gen=50 **cross=0.8**
1 HM	50	71.4 90 55.2	0.49 0.84 0.90	38 41 88	2.34	1102	8.0112	[71.9, 94.7, 59, 0.5, 0.83, 19.8]
2 GA	18.8	45.9 63 62.3	0.40 0.80 0.85	382 395 400	2.81	10391	9.9733	dim=50 gen=15 **cross =0.6**
2 HM	10	69.2 90 58.6	0.45 0.85 0.91	79 83 129	2.42	3239	8.1261	[45.9, 63, 62.3, 0.4, 0.8, 18.8]
3 GA	14.2	62.8 84.8 35.6	0.46 0.84 0.90	110 113 400	2.40	4152	8.2444	dim=20 gen=50 **cross =1**
3 HM	10	69.2 90 58.6	0.45 0.85 0.91	79 83 129	2.42	3239	8.1261	62.8, 84.8, 35.6, 0.46, 0.84, 14.2]

Concerning the two variants of crossover and mutation, the adequate method depends on the system to be optimized and on the other parameters established for GA procedure. In our case, no significant differences are between the two crossover or mutation variants (Table 4).

Another parameter of GA to be tested is the value of mutation (*mut*). Table 5 shows that a small value for this parameter is a good choice for the optimization (*mut* = 0.01). Generally, good results are obtained with HM procedure.

Different selection methods are tested and the results are presented in table 6. The best method seems to be rank selection and HM becomes useful in the other two: roulette and tournament selection.

Finally, the complete objective function noted (2) is used in optimization along with different variants of GA. HM is a good optimization method (optimization 1 and 2 in table 7), except the case where the initial values for SQP (within HM) are not adequate for the convergence (optimization 3 and 4 in table 7). Also, the GA parameters represents another decision factor for the optimization method.

Table 4

The influence of the crossover variants upon optimization results

No.	I_0 (mol/m^3)	T (^{o}C)	x	t (min)	Q	DP_n	Objective function	Observations
1 GA	20.5	59 81.6 48.3	0.44 0.82 0.88	126 131 179	2.43	3837	8.5060	dim=20 gen=30 cross=0.8; **crossover 1**
1 HM	50	71.4 90 55.2	0.49 0.84 0.90	38 41 88	2.34	1102	8.0112	[59, 81.6, 48.3, 0.44, 0.82, 20.5] w_Q=3; w_x=10
2 GA	27.6	101.3 57.8 48	0.88 0.95 0.97	11 400 400	3.19	443	9.8602	dim=20 gen=30 cross=0.8; **crossover 2**
2 HM	28.3	90 57.8 47.7	0.88 0.89 0.95	19 20 500	3.44	761	10.8321	[101.3, 57.8, 48, 0.88, 0.95, 27.6] w_Q=3; w_x=10

Table 5

The influence of the mutation rate upon optimization results

No.	I_0 (mol/m^3)	T (^{o}C)	x	t (min)	Q	DP_n	Objective function	Observations
1 GA	15.6	61.4 85 36.5	0.46 0.85 0.90	117 122 400	2.41	3961	8.2266	dim=20; gen=30 w_x=10; cross=0.8 **mut=0.01**
1 HM	15.6	61.4 85 40	0.44 0.85 0.92	116 121 500	2.43	3677	8.1200	[61.4, 85, 36.5, 0.46, 0.85, 15.6] w_Q=3; w_x=10
2 GA	24.7	55.8 77.1 56.8	0.38 0.81 0.91	146 154 209	2.74	3546	9.1390	dim=20; gen=30 w_x=10; cross=0.8 **mut=0.03**
2 HM	24.7	55.8 77.1 56.8	0.38 0.81 0.91	146 154 209	2.74	3546	9.1390	[55.8, 77.1, 56.8, 0.38, 0.81, 24.7] w_Q=3; w_x=10
3 GA	35.2	74.1 91 39.8	0.43 0.85 0.92	34 38 400	2.46	1145	9.0733	dim=20; gen=30 w_x=20; cross=0.8 **mut=0.03**
3 HM	50	71.5 90 40	0.49 0.86 0.92	38 41 500	2.34	1098	7.7898	[74.1, 91, 39.8, 0.43, 0.85, 35.2] w_Q=3; w_x=10
4 GA	20.5	62.7 86.2 37.7	0.42 0.85 0.91	94 99 400	2.43	2674	10.1231	dim=20; gen=30 w_x=20; cross=0.8 **mut=0.03**
4 HM	20.5	62.7 86.2 40	0.42 0.85 0.92	94 100 500	2.45	2666	8.1678	[62.7, 86.2, 37.7, 0.42, 0.85, 20.5] w_Q=3; w_x=10
5 GA	17.3	65.1 87.5 46.9	0.41 0.81 0.91	84 89 143	2.58	2587	9.5122	dim=20; gen=30 w_x=20; cross=0.8 **mut=0.04**
5 HM	17.3	65.1 87.5 46.9	0.41 0.81 0.87	83 88 140	2.53	2702	8.8497	[65.1, 87.5, 46.9, 0.41, 0.81, 17.3] w_Q=3; w_x=10

Table 6

Different selection variants used within the GA optimization method

No.	I_0 (mol/m^3)	T (^{o}C)	x	t (min)	Q	DP_n	Objective function	Observations
1 GA	12	94.6 96.1 114	0.33 0.41 0.997	12 14 32	2.65	608	8.0170	dim=20; gen=30; w_x=20 cross=0.8; mut=0.05 **rank selection**
1 HM	10	80.6 90 90	0.40 0.60 0.98	35 41 74	4.55	1174	13.812	[94.6, 96.1, 114, 0.33, 0.41, 12] w_Q=3; w_x=10
1' GA	12.9	60.7 80.9 46.2	0.42 0.81 0.87	129 135 184	2.45	4961	9.9291	dim=20; gen=30; w_x=20 cross=0.8; mut=0.05 **roulette selection**
1' HM	12.9	60.7 80.9 46.2	0.42 0.81 0.87	129 135 184	2.45	4961	9.9291	[60.7, 80.9, 46.2, 0.42, 0.81, 12.9] w_Q=3; w_x=10
1'' GA	29.9	76.9 82.5 55.2	0.46 0.85 0.90	31 36 83	3.14	1454	11.3919	dim=20; gen=30; w_x=20 cross=0.8; mutație=0.05 **tournament selection**
1'' HM	50	71.4 90 55.2	0.49 0.84 0.90	38 41 88	2.34	1102	8.0112	[76.9, 82.5, 55.2, 0.46, 0.85, 29.9] w_Q=3; w_x=10

Table 7

Optimizations performed with a complete objective function

No.	I_0 (mol/m^3)	T (^{o}C)	x	t (min)	Q	DP_n	Objective function	Observations
1 GA	27.7	78.4 51.3 65.1	0.85 0.94 0.95	35 400 400	4.27	1364	14.4744	dim=20 ; gen=30; w_x=20; w_Q=3; w_t=10^{-5}; DP_{nd}=1500; cross=0.8 ; mut=0.05; rank selection
1 HM	10	90 50.3 53.1	0.90 0.98 0.98	29 500 500	3.62	1469	12.0431	[78.4, 51.3, 65.1, 0.85, 0.94, 27.7]
2 GA	38.3	59.6 67.8 70	0.36 0.44 0.95	88 94 151	5.81	1522	18.5963	dim=20 ; gen=30; w_x=20; w_Q=3; w_t=10^{-5}; DP_{nd}=1500; cross=0.8 ; mut=0.05; rank selection
2 HM	10	71.9 49.8 85.3	0.25 0.25 0.98	45 45 98	4.87	1463	15.1554	[59.6, 67.8, 70, 0.36, 0.44, 38.3]
3 GA	41.9	42.8 76.1 76.6	0.37 0.50 0.93	374 379 400	5.28	1680	18.1458	dim=20 ; gen=30; w_x=20; w_Q=3; w_t=10^{-5}; DP_{nd}=1500; cross=0.8 ; mut=0.05; roulette selection
3 HM	41.2	43.6 75.9 73.6	0.1 0.73 0.96	109 136 185	6.48	907	28.2602	[42.8, 76.1, 76.6, 0.37, 0.50, 41.9]
4 GA	14.4	40.7 59.7 72.4	0.12 0.40 0.97	279 362 400	6.29	1660	20.3103	dim=20 ; gen=30; w_x=20; w_Q=3; w_t=10^{-5}; DP_{nd}=1500; cross=0.8 ; mut=0.05; tournament selection
4 HM	28.1	41.3 62.6 89.3	0.1 0.1 0.97	163 500 500	12.13	1451	37.3112	[40.7, 59.7, 72.4, 0.12, 0.40, 14.4]

One obvious problem of the optimization that combines objectives into a single function is that it may be difficult to generate a set of weights (namely *w* in equation 2) 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. GA designed in this paper presents the advantage of computing the optimal values for the weights of the objectives, along with the optimal values for decision variables by coding them into chromosomes.

To illustrate this aspect, the objective function (2) with $w_t = 0$ and the decision variables I_0 and T (one isothermal step of temperature) were considered. Table 8 contains the parameters of the GA method, the optimal values of the weights and decision variables calculated within GA, the results of the kinetic model obtained with optimal values I_0 and T and, in the last column, the value of the objective function and the imposed DP_n (DP_{nd}).

Table 8

Optimizations were GA computes optimal values for decision variables and weight coefficients

No.	GA method	Computed weights	Decision variables	Outputs of the model	Objective function
1	dim = 20; gen = 30 cross = 0.8; mut = 0.05 rank selection	w_x = 22.46 w_Q = 10.25 w_{DPn} = 18.61	T = 89.57 I_0 = 12.95	x = 0.98; Q = 5.27 DP_n = 848.14 DP_w = 4474.23	J = 54.543158 DP_{nd} = 800
2	dim = 20; gen = 30 cross = 0.8; mut = 0.05 rank selection	w_x = 16.38 w_Q = 10.41 w_{DPn} = 264.90	T = 88.62 I_0 = 10.92	x = 0.98; Q = 5.21 DP_n = 1001.49 DP_w = 5223.63	J = 54.611938 DP_{nd} = 1000
3	dim = 20; gen = 30 cross = 0.8; mut = 0.05 rank selection	w_x = 27.66 w_Q = 10.38 w_{DPn} = 58.14	T = 81.72 I_0 = 11.07	x = 0.97; Q = 5.28 DP_n = 1498.85 DP_w = 7915.11	J = 55.670337 DP_{nd} = 1500
4	dim = 20; gen = 30 cross = 0.8; mut = 0.05 rank selection	w_x = 19.35 w_Q = 11.35 w_{DPn} = 68.51	T = 81.46 I_0 = 10.56	x = 0.97; Q = 5.25 DP_n = 1568.80 DP_w = 8246.59	J = 61.397333 DP_{nd} = 1800

In this paper, the optimization procedure applied to MMA polymerization process intends to illustrate several aspects: the importance of proper selection for GA operators and parameters, the efficiency of HM optimization technique, the possibility to handle a scalar multiobjective function with weight coefficients computed within the GA method. To obtain automatically the optimal values for the weights is an advantage when little is known a priori about the process. On the other hand, if the user establishes these values before running the optimization procedure, priority can be given to one of the individual objectives.

In the above tables, the best optimization corresponds to the minimum objective function from the mathematical point of view, or, based on technological considerations, to some individual criterion accomplished priority.

Our future work intends to combine the optimization techniques based on genetic algorithm with a neural network model, avoiding the difficulties of designing and handling an accurate process model.

CONCLUSIONS

In this study, we have demonstrated that a hybrid search procedure that employs a genetic algorithm followed by a local optimizer can identify the global optimum, in reasonable amounts of computer time. This shows that the hybrid search algorithm is a powerful tool for solving realistic complex optimization problems. Besides the time efficiency, this hybrid procedure has the advantage of not relying on any specific mathematical structure of the underlying problem, which makes its implementation much simpler.

A simple GA solves the optimization problem, also providing the values of the objective weights beside the optimal values of the decision variables, in order to minimize a scalar objective function. Although the problem of choosing the appropriate weights in the scalar objective function seems to be a difficult task, computing the optimal values for the weights within the GA is a good attempt to provide a solution to this problem.

Acknowledgements: This research occurred in the framework of the Project PN II PC 71-006/18.09.2007 for which the authors acknowledge financing authority.

REFERENCES

1. S. Curteanu, *Bulletin of IPI*, **2006**, *L* (*LIV*), 77-90.
2. F. Leon and S. Curteanu, *Mater. Plast.*, **2007**, *44*, 211-219.
3. H. Halsall-Whitney and H. Thibault, *Comp. Chem. Eng.*, **2006**, *30*, 1155-1168.
4. S.S. Chakravarthy, D.N. Saraf and S.K. Gupta, *J. Appl. Polym. Sci.*, **1997**, *63*, 529-548.
5. S. Garg, S.K. Gupta and D.N. Saraf, *J. Appl. Polym. Sci.*, **1999**, *71*, 2101-2120.
6. F. Zhou, S.K. Gupta and A.K. Ray, *J. Appl. Polym. Sci.*, **2000**, *78*, 1439-1458.
7. B.V. Babu and R. Angira, *Comp. Chem. Eng.*, **2006**, *30*, 989-1002.
8. J. Grefensetette, *Proceedings of the Fourth International Conference on Genetic Algorithms*, San Mateo, CA: Morgan Kaufmann Publishers, **1994**.
9. P. Moscato and M. Norman, *Proceedings of the International Conference on Parallel Computing and Transportation Applications*, **1992**.
10. M. Okamoto, T. Nonasaka, S. Ochiai and D. Tominaga, *Appl. Math. Comput.*, **1998**, *91*, 63-72.
11. G.P. Kasprzyk and M. Jasku, *J. Electroanalytical Chem.*, **2004**, *567*, 39-66.
12. S.A. Torabi, S.M.T. Fatemi Ghomi and B. Karimi, *Eur. J. Oper. Res.*, **2006**, *173*, 173-189.
13. L. Wang, L. Zhang and D.Z. Zheng, *Comp. Operat. Res.,* **2006***, 33*, 2969-2971.
14. H.F. Wang and K.Y. Wu, *Comp. Operat. Res.,* **2004**, *31*, 2453-2471.
15. W. Ho and P. Ji, *Comp. Operat. Res.,***2003**, *30*, 2175-2189.
16. C.F. Liaw, *Eur. J. Oper. Res.*, **2000**, *124*, 28-42.
17. S. Curteanu and V. Bulacovschi, *Roum. Chem. Quart. Rev.*, **1999**, *7*, 281-296.
18. S. Curteanu, V. Bulacovschi and M. Constantinescu, *Hung. J. Ind. Chem., **1999**, 27,* 287-293.