Multiobjective optimization of a free radical bulk polymerization reactor using genetic algorithm

*Sanjeev Garg, Santosh K. Gupta**[a]

Department of Chemical Engineering, Indian Institute of Technology, Kanpur 208 016, India
skgupta@iitk.ernet.in

(Received: June 3, 1998; revised: August 17, 1998)

SUMMARY: A multiobjective optimization technique has been developed for free radical bulk polymerization reactors using genetic algorithm. The polymerization of methyl methacrylate in a batch reactor has been studied as an example. The two objective functions which are minimized are the total reaction time and the polydispersity index of the polymer product. Simultaneously, end-point constraints are incorporated to attain desired values of the monomer conversion (x_m) and the number average chain length (μ_n). A nondominated sorting genetic algorithm (NSGA) has been adapted to obtain the optimal control variable (temperature) history. It has been shown that the optimal solution converges to a unique point and no Pareto set is obtained. It has been observed that the optimal solution obtained using the NSGA for multiobjective function optimization compares very well with the solution obtained using the simple genetic algorithm (SGA) for a single objective function optimization problem, in which only the total reaction time is minimized and the two end-point constraints on x_m and μ_n are satisfied.

Introduction

The physical properties of any polymer depend largely on its molecular weight distribution (MWD). From the point of view of applications, it is desirable to have a high molecular weight product, often with a narrow MWD (at times, though, broad MWD products may be more desirable). Martin et al.[1] and Nunes et al.[2] showed that narrowing the MWD improves the thermal properties, stress-strain relationships, impact resistance, hardness and strength of the polymer. In order to produce such materials in industrial polymerization reactors, we must have appropriate (optimal) operating conditions. Several studies have been reported on the optimization of polymerization reactors, and these have been reviewed recently by Farber[3], Chakravarthy et al.[4] and Mitra et al.[5] A detailed discussion is, therefore, not being provided here. The thrust in the recent past has been on the optimization of reactors using multiple objective functions, as well as end-point (product) constraints. It is found that the operating variables for polymerization reactors influence the properties of the product in conflicting ways, i.e., these complex systems are such that any desirable change in one objective function often leads to a detrimental change in another objective function. A multi-criteria analysis for optimization of the operating conditions is, therefore, quite important for industrial reactors.

In the present study, we discuss the multiobjective optimization of a free-radical bulk polymerization batch reactor which produces a desired end-product. The example system chosen is the polymerization of methyl methacrylate (MMA). An important objective function for this system is the minimization of the final reaction time, t_f. This leads to higher productivity. The other objective function is the minimization of the polydispersity index (PDI or Q) of the polymer product. This ensures good physical properties of the polymer manufactured. Two end-point constraints are also used. The first is to attain a desired (high) value of the monomer conversion, x_m. This ensures low post-reactor processing costs. The other important end-point constraint is to produce polymer having a desired value of the number average chain length, μ_n. The temperature history, $T(t)$, alone [rather than $T(t)$ as well as an initiator addition history] is used as the 'control' or operating variable. Sacks et al.[6] studied a similar optimization problem (but with a single objective function only involving a weighted average of these individual objectives) over twenty five years ago, using the early models for the Trommsdorff[7] effect and using relatively elementary optimization techniques. With more powerful and robust optimization techniques now available[4,8–12] which can solve problems involving multiple objectives and end-point constraints correctly, it is appropriate that these be applied to study such problems, and see if similar solutions are obtained or not.

This study accounts for the gel (or Trommsdorff) effect[7] which is exhibited during the polymerization of MMA. The free-volume theory of Vrentas and Duda[13] is used to model this manifestation of diffusional resistance.

[a] On leave to: The Department of Chemical Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260. Email: cheskg@nus.edu.sg.

  1022-1344/99/0101–0046$17.50+.50/0

The detailed theory and the rate constants and parameters required for describing the Trommsdorff[7] effect are given in our previous papers[4, 14, 15], and are not being repeated here. An adapted version[5] (known as Nondominated Sorting Genetic Algorithm, NSGA) of genetic algorithm[8, 9], an artificial intelligence based technique, is used for obtaining optimal temperature histories. Mitra et al.[5] have used this technique for the optimization of an industrial nylon 6 reactor. By formulating a multi-objective optimization problem for the PMMA batch reactor we intend to investigate quantitatively the trade offs among individual objectives and try to generate a wider range of alternative operating policies.

Formulation

The mathematical model for the present study is based on the general free radical polymerization kinetic scheme shown in Tab. 1. It consists mainly of four major steps: initiation, propagation, termination and chain transfer. The mass balance and moment equations for methyl methacrylate (MMA) polymerization in a semi-batch reactor can easily be written, based on the kinetic scheme, and are reported elsewhere[4, 12, 14]. These are not reproduced here for the sake of brevity. Similarly, the values of the different parameters to integrate the model equations are also extensively reported[4, 12, 14]. In general, the state variable equations can be written in the form:

$$\mathrm{d}\boldsymbol{x}/\mathrm{d}t = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}); \qquad \boldsymbol{x}\,(t = 0) = \boldsymbol{x}_0 \tag{1}$$

where $\boldsymbol{x}$ is the state variable vector given by

$$\boldsymbol{x} = [I, M, R, \lambda_0, \lambda_1, \lambda_2, \mu_0, \mu_1, \mu_2, \zeta_{m1}]^{\mathrm{T}} \tag{2}$$

and $\boldsymbol{u}$ is the control variable vector (here it is a scalar)

Tab. 1. Kinetic scheme for addition polymerization

Initiation	$I \xrightarrow{k_d} 2R$
	$R + M \xrightarrow{k_i} P_1$
Propagation	$P_n + M \xrightarrow{k_p} P_{n+1}$
Termination by combination	$P_n + P_m \xrightarrow{k_{tc}} D_{n+m}$
Termination by disproportionation	$P_n + P_m \xrightarrow{k_{td}} D_n + D_m$
Chain transfer to monomer	$P_n + M \xrightarrow{k_f} P_1 + D_n$
Chain transfer to monomer	$P_n + S \xrightarrow{k_s} S + D_n$
via solvent	$S + M \xrightarrow{fast} S + P_1$
	or
	$S + M \xrightarrow{k_s} D_n + P_1$

k_{tc}, k_f and k_s are taken as zero in the present study (bulk polymerization of MMA)

$$\boldsymbol{u}\,(t) = u(t) = T(t) \tag{3}$$

The initial value problem (IVP) given by the ordinary differential equations (ODEs) in Eq. (1) can be integrated using the D02EJF subroutine of the NAG library for any given $T(t)$ and initial values of the state variables. This subroutine uses Gear's technique[16] for integrating a set of stiff ODEs. Provision was made in the algorithm for self-adjustment of the error-tolerance (*TOL* in the code).

The monomer conversion in a semi-batch reactor at any time, t, is defined as

$$x_m = (1 - M/\zeta_{m1}) \tag{4}$$

where ζ_{m1} is the total monomer added till time t. The multiobjective function optimization problem studied in this work is

$$\text{Min } \boldsymbol{I}[T(t)] \equiv [I_1, I_2]^{\mathrm{T}} = [t_f, Q_f]^{\mathrm{T}} \tag{5a}$$

subject to

$$x_{mf} = x_{md} \pm TOL_1 \tag{5b}$$

$$\mu_{\mathrm{nf}} = \mu_{\mathrm{nd}} \pm TOL_2 \tag{5c}$$

$$\mathrm{d}\boldsymbol{x}/\mathrm{d}t = \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}) \qquad \boldsymbol{x}\,(t = 0) = \boldsymbol{x}_0 \tag{5d}$$

$$T_{min} \leq T \leq T_{max} \tag{5e}$$

where x_{md} and μ_{nd} are the desired values of the monomer conversion and of the number average chain length at the final (total reaction) time, t_f. TOL_1 and TOL_2 are taken as 0.005 and 20.00, respectively. The constraints on the values, x_{mf} and μ_{nf}, are incorporated in the first objective function, I_1, in the form of penalty functions with the (large) weightage factors, w_1 and w_2

$$I_1 = t_f + w_1\left(1 - \frac{x_{mf}}{x_{md}}\right)^2 + w_2\left(1 - \frac{\mu_{nf}}{\mu_{nd}}\right)^2 \tag{6}$$

Minimization of I_1 leads to an increase in the production capacity through a reduction of t_f, while simultaneously giving preference to solutions satisfying the end requirements. The second objective function, I_2, also incorporates the violations of x_{mf} and μ_{nf} from their desired values with the same weightage factors, w_1 and w_2

$$I_2 = Q_f + w_1\left(1 - \frac{x_{mf}}{x_{md}}\right)^2 + w_2\left(1 - \frac{\mu_{nf}}{\mu_{nd}}\right)^2 \tag{7}$$

Minimization of Q_f ensures a narrow breadth of the molecular weight distribution of the product, and, hence, product with desired physical properties, while simultaneously satisfying the end-point requirements on x_m and μ_n.

The two objective functions are *usually* conflicting in nature and, therefore, provide an excellent opportunity for carrying out multiobjective function optimization. The solution of the multiobjective optimization problem

described in Eq. (5) is obtained using the nondominated sorting genetic algorithm (NSGA[8–10]) adapted by our group so as to apply for control variables which are continuous functions of time. Details of this adapted NSGA are given elsewhere[5]. The Appendix provides a short introduction. The optimization technique usually (but not always) leads to several feasible solutions which satisfy the end-point constraints on x_{mf} and μ_{nf} (within specified values of the tolerances). These solutions form what is referred to as a Pareto set. The latter comprises several non-inferior or nondominated (optimal) points which have the characteristic that if we go from one point to another on this set, one objective function improves but the other worsens.

Results and discussion

The adapted NSGA code for multiobjective optimization was run for a PMMA batch reactor for the following case (referred to as the reference case)

$$x_{md} = 0.94\ (\pm 0.005) \tag{8a}$$

$$\mu_{nd} = 1850\ (\pm 20) \tag{8b}$$

$$60\,^{\circ}\mathrm{C} \leq T(t) \leq 90\,^{\circ}\mathrm{C} \tag{8c}$$

These values are quite close to those used by Vaid and Gupta[17] and Chakravarthy et al.[4], and are identical to those used by Garg et al.[12] for single objective function studies. The same values of x_{md} and μ_{nd} are being used here so that we can compare the results for the single and the double objective function optimizations. The computer code developed herein was tested using standard checks[5, 12]. The CPU time required for running the program for about 30 generations on a DECα 1000 was 69 s. Tab. 2 gives the values of the several parameters used in this work.

Fig. 1 shows all the feasible solutions (i.e., those satisfying the end-point constraints in Eqs. (8a) and (8b) at different values of N_g, the generation number. In the initial generations, the feasible points move around in the $t_f - Q_f$ space, but as the generations evolve, the feasible points move towards a unique point. It may be added that there are 61 feasible points in Fig. 1 (N_g = 10 and 30), several being duplicates or being almost indistinguishable from the points shown. The points are shown for generations 15 to 30 in a highly expanded scale in Fig. 2. The solution of our multiobjective function optimization problem with end-point constraints (Eq. (5)), thus, is a unique point (taken as point A in Fig. 2) with

$$t_f = 1696.96\ \mathrm{s} \qquad x_{mf} = 0.9396$$
$$Q_f = 11.6275 \qquad \mu_{nf} = 1850.58 \tag{9}$$

Point A has been selected from among the several points in Fig. 2 such that the end-point values of x_m and

Tab. 2. Computational parameters used in this study[a]

$N_{chr} = 70$	$q = 15$
$N_{ga} = 10$	$TOL_1 = 0.005$
$N_{g,\,max} = 30$	$TOL_2 = 20.00$
$N_P = 100$	$T_k^{\max} = +15.0\,^{\circ}\mathrm{C}$; $k = 1, 2, ..., N_{ga}$
$N_{sim} = 100$	$T_k^{\min} = -15.0\,^{\circ}\mathrm{C}$; $k = 1, 2, ..., N_{ga}$
$N_{str} = 7$	$t_{f_0} = 4000$ s
$p_c = 0.99$	$w_1 = w_2 = 0.25 \times 10^6$
$p_m = 0.0$	$\alpha = 2$

[a] Parameters for NSGA described in the Appendix as well as in ref.[5]

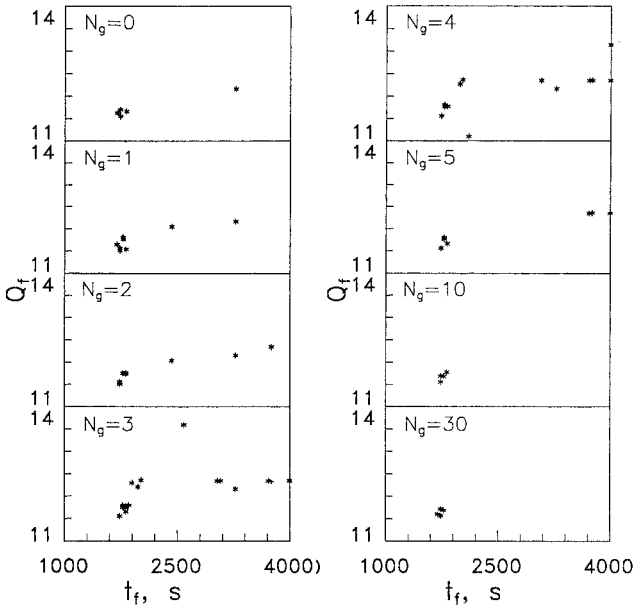

Fig. 1. Q_f vs. t_f space showing convergence of the feasible solutions over the generations. For N_g = 10 and 30, 61 final points were obtained (N_p = 100), x_{md} = 0.94, μ_{nd} = 1850

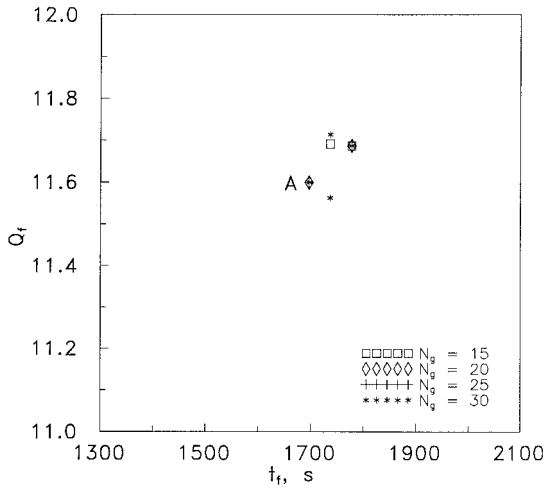

Fig. 2. Feasible solutions over the generations with expanded scales. Point **A** is taken as the optimal solution

Tab. 3. Effect of different $[I]_0$, x_{md} and μ_{nd} values on the optimal solution

$\frac{[I]_0}{\text{mol}\cdot\text{m}^{-3}}$	x_{md}	μ_{nd}	x_{mf}	μ_{nf}	t_f/s	Q_f
15.48[a)]	0.94	1850	0.9397	1850.6	1697	11.63
15.48	0.94	1900	0.8951	1900.8	1697	10.40
15.48	0.92	1850	0.9186	1860.3	1616	11.11
25.80	0.94	1800	0.9409	1798.6	2182	9.64
25.80	0.92	1800	0.9208	1801.4	2020	9.94
25.80	0.94	1850	0.9401	1848.0	2626	10.48

[a)] Reference case.

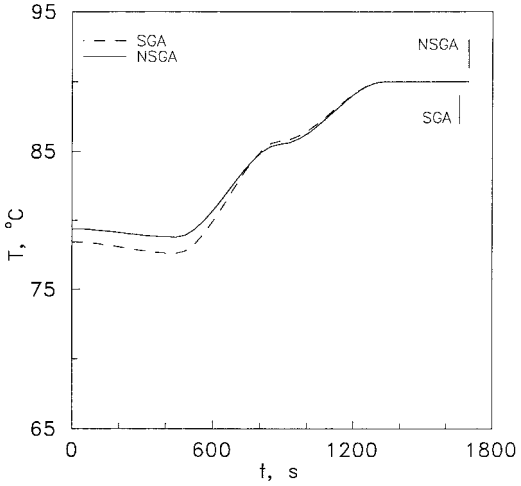

Fig. 3. Temperature history for the optimal solution (point **A** in Fig. 2). The SGA history is also shown for comparision

μ_n lie closest to the desired values. No Pareto set of non-inferior solutions is obtained in this study. A similar conclusion (unique solution) is obtained for other combinations of x_{md} and μ_{nd}, as observed from Tab. 3.

The optimal temperature history, $T_{opt}(t)$, for the optimal solution described in Eq. (9) (reference case) is shown in Fig. 3. It may be emphasized that in this study, a heat-balance equation is not used. Hence, the optimal temperature histories, $T_{opt}(t)$, in the well-stirred batch reactor must be *implemented* using appropriate computer-control. This may be relatively easy to achieve (using heating from a jacket fluid), at least during the early stages of reaction, when the temperature of the reaction mass goes up relatively slowly. However, trying to maintain the temperature of the reaction mass at $T_{opt}(t)$ during the later stages of reaction when the Trommsdorff effect manifests itself, may not be as trivial, because of the excessive amounts of the exothermic heat of reaction released in a relatively short period of time. The focus of this study is to compute $T_{opt}(t)$, and not the constraints associated with its industrial implementation.

The optimal temperature history corresponding to our previous[12)] optimization study using a *single objective function only*, and described by

$$\text{Min } I\,[T(t)] \equiv t_f \tag{10a}$$

subject to

$$x_{mf} = x_{md} = 0.94 \tag{10b}$$

$$\mu_{nf} = \mu_{nd} = 1850 \tag{10c}$$

$$\mathrm{d}\boldsymbol{x}/\mathrm{d}t = \boldsymbol{f}(\boldsymbol{x}, T) \qquad \boldsymbol{x}\,(t = 0) = \boldsymbol{x}_0 \tag{10d}$$

$$60\,^{\circ}\mathrm{C} \leq T \leq 90\,^{\circ}\mathrm{C} \tag{10e}$$

is also shown in Fig. 3. The (unique) optimal solution for Eq. (10) was found using the simple GA (SGA)[8, 9)], as

$$\begin{aligned} t_f &= 1656.56 \text{ s} \qquad x_{mf} = 0.9354 \\ Q_f &= 11.2323 \qquad \mu_{nf} = 1854.99 \end{aligned} \tag{11}$$

The end-point values of x_{mf} and μ_{nf} in Eq. (11) are quite similar to those in Eq. (9). It is observed that the two temperature histories in Fig. 3 are more or less the same, except for some deviation in the early stages of reaction. Fig. 4–6 show the variations of x_m, μ_n and Q with time for the optimal solutions for both the single and multiobjective cases (Eqs. (5) and (7)). Again, the variations for the two optimization problems are not too significant. It may be noted that the value of Q shoots up from about 2.0 to about 11.6 in an extremely short period of time during which the Trommsdorff effect manifests itself. The high value of Q is because of the production of material having very high molecular weights (instantaneous values) during this short interval of time (essentially, a bimodal molecular weight distribution is expected for the final product).

The two interesting results from the present study, thus, are

(i) the uniqueness of the optimal solution of the multiobjective optimization problem (Eq. (5)), and
(ii) the fact that the single optimal solution obtained for the two-objective-function problem is similar to that obtained in the single objective function problem (Eq. (10)).

In the latter problem, the polydispersity index of the product is not considered. A considerable amount of controversy has existed in the open literature on the effect of incorporating the PDI in the optimization problem using somewhat indirect approaches[18)] in which an objective function is used which incorporates several objectives with questionable values of weightage factors. These approaches suffer from a drawback[19, 20)] that certain optimal solutions can be missed, irrespective of the values of the weightage factors associated with the several individual objectives. This happens if the non-convexity of the objective function gives rise to a duality gap. To the best of our knowledge, this is the first time that a *formally* correct procedure has been used to solve a problem incorporating objective functions involving the minimization of t_f

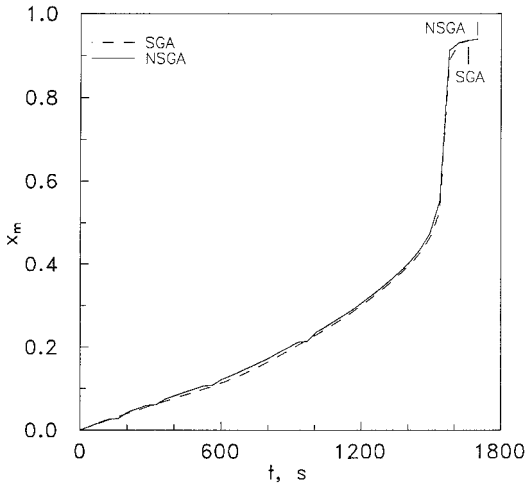

Fig. 4. Variation of monomer conversion with time for the optimal temperature histories shown in Fig. 3

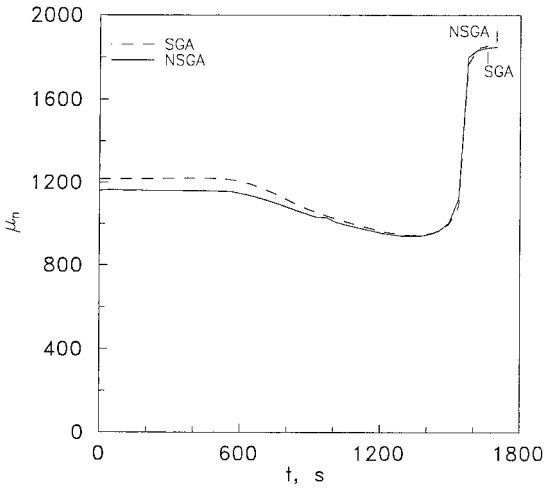

Fig. 5. Variation of the number average chain length with time for the temperature histories shown in Fig. 3

and Q_f, while ensuring desired values of x_{mf} and μ_{nf}. It may be pointed out that Chakravarthy et al.[4] had found some *indications* in their study that minimization of t_f alone *possibly* ensures the minimization of Q_f in the presence of end-point constraints on x_{mf} and μ_{nf}. They had *not*, however, actually solved the problem involving the minimization of both t_f and Q_f.

Conclusions

A unique solution has been obtained for the multiobjective optimization problem described in Eq. (5). No Pareto set has been obtained. Temperature history has been used as the control variable while minimizing Q_f as well as t_f, while constraining x_{mf} and μ_{nf} to lie at desired values. An adapted NSGA technique has been used to obtain the optimal solution. This is the first time in the open literature that a rigorously correct procedure has been used to study this interesting multiobjective optimization problem in polymer reaction engineering.

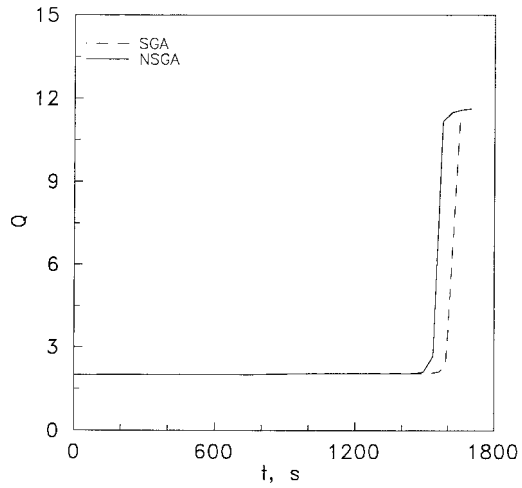

Fig. 6. Variation of the polydispersity index with time for the temperature histories shown in Fig. 3

Nomenclature

D_n	dead polymer molecule having n repeat units
$\boldsymbol{I}$	vector of objective functions
I	moles of initiator at any time t (in mol)
I_1, I_2	objective functions
k_d, k_p, k_t	rate constants for initiation, propagation and termination in presence of the gel and glass effects (in s^{-1}, or $m^3 \cdot mol^{-1} \cdot s^{-1}$)
M	moles of monomer in the liquid phase (in mol)
N_{chr}	total number of binary digits in chromosome
N_g	generation number
N_{ga}	number of u values which GA generates
N_p	number of chromosomes
N_{sim}	number of u values after interpolation
N_{str}	number of binary digits representing each of the control variable values
P_n	growing polymer radical having n repeat units
p_c	crossover probability
p_m	mutation probability
Q	polydispersity index $\left[\equiv \dfrac{(\lambda_2+\mu_2)(\lambda_0+\mu_0)}{(\lambda_1+\mu_1)^2}\right]$
q	desired number (approx.) of Pareto points required to be generated
R	moles of primary radical; universal gas constant (in $atm \cdot m^3 \cdot mol^{-1} \cdot K^{-1}$)
T	temperature of the reaction mixture at time t (in K)
TOL_1, TOL_2	allowed tolerances on x_{mf} and μ_{nf}, respectively
t	time (in s)
t_f	final reaction time (in s)
$\boldsymbol{u}$	control vector (scalar, u, in this work)
w_1, w_2	weightage factors

$\boldsymbol{x}$ vector representing state variables
x_m monomer conversion (molar) at time t

Greek letters

α exponent controlling the sharing effect
λ_k kth (k = 0, 1, 2, ...) moment of live polymer radicals (P_n)

$$\left[\equiv \sum_{n=1}^{\infty} n^k P_n\right] \text{ (in mol)}$$

μ_k kth (k = 0, 1, 2, ...) moment of dead (D_n) polymer chains

$$\left[\equiv \sum_{n=1}^{\infty} n^k D_n\right] \text{ (in mol)}$$

μ_n number average chain length at time t
ζ_{m1} net monomer added till time t

Subscripts/Superscripts

d desired values
f final values (at $t = t_f$)
max maximum value
min minimum value
0 initial value (at time t = 0)

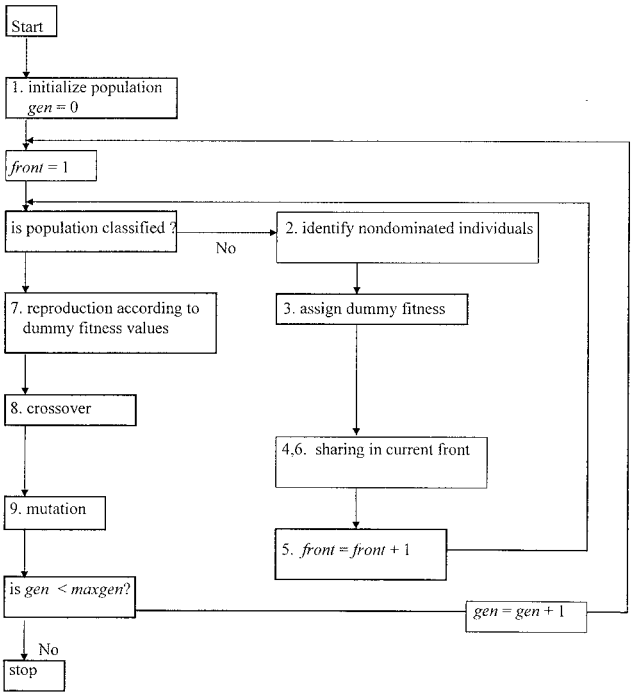

Fig. 7. A flowchart of the adapted NSGA. The numbers in some of the boxes correspond to the sections in the Appendix

Appendix: Details of the nondominated sorting genetic algorithm[5)]

Fig. 7 shows a flowchart of the algorithm (NSGA) used in this work. More details are provided in ref.[5)]

1. At generation number, N_g = 0, a *population* having N_p members (called chromosomes) is generated (initialize population in Fig. 7). Each chromosome in the population carries the information of one digitized control variable *history* [digitized set of values of the temperature, $u(t) \equiv T(t)$]. We discretize our control variable history, $T(t)$, in terms of N_{ga} equispaced points in $0 \le t \le t_{f0}$ (t_{f0}, an initial estimate of t_f, is to be supplied). Thus, each of the N_p chromosomes (called strings) comprises of a sequence of N_{ga} numbers (called substrings). Each of these substrings, in turn, comprises a set of N_{str} binary numbers (0 or 1). Each chromosome, therefore, has $N_{chr} = N_{ga} \times N_{str}$ binary digits. The N_{chr} individual binaries in each of the N_p chromosomes are generated using a random number generation subroutine.

The complete binary string (sequence of N_{chr} binaries) of the i^{th} chromosome, when decoded into real numbers, $u_{P,k}^{(i)}$, and interpolated (mapped) between the upper ($u \le u^{max}$) and lower ($u \ge u^{min}$) bounds of the control variables, u, at that location, gives a digitized $u_P^{(i)}$-history (a set of N_{ga} *real* values), $[u_{P,1}^{(i)}, u_{P,2}^{(i)}, ..., u_{P,N_{ga}}^{(i)}]$, representing a $T(t)$ history. Thus, there is a set of N_p chromosomes in the initial population, each representing a digitized $u_P(t)$ [$\equiv T(t)$] history, and each appropriately coded in the form of a string of N_{chr} binaries.

The decoded and adaptively mapped discretized values, $u_{P,k}^{(i)}$, are curve-fitted piece-wise (splines) to obtain a continuous *function*, $U_P^{(i)}(t)$. A piece-wise cubic Hermite subroutine (E01BFF from the NAG library) is used to do this. This continuous function is again digitized to give N_{sim} ($\ge N_{ga}$) values of the control variable, [$U_{P,1}^{(i)}$; l = 1, 2,..., N_{sim}].

These more closely spaced, discretized values of $U_P^{(i)}(t)$ are fed to the simulation package, D02EJF (of NAG library) which integrates the state variable equations (Eq. (1)) starting with the given initial conditions and terminating at the stopping condition, $x_{mf} = x_{md}$. The values of the two objective functions, $I_1^{(i)}$ and $I_2^{(i)}$ [at the final reaction time $t = t_f$, see Eqs. (6) and (7)], are computed for all the N_p chromosomes.

One additional point needs to be emphasized. The computer codes involving GA usually *maximize* a fitness function, $F_m^{(i)}$, rather than minimize objective functions, $I_m^{(i)}$, m = 1, 2. Hence, we define fitness functions to convert the minimization problem to an equivalent maximization problem as follows:

$$F_1^{(i)} \equiv 1/(1 + I_1^{(i)})$$
$$F_2^{(i)} \equiv 1/(1 + I_2^{(i)}) \qquad (12)$$

All the *feasible* points from among the N_p chromosomes are identified (for plotting) at this stage. The feasible points or chromosomes are those satisfying Eqs. (5b) and (5c).

2. A chromosome, i_1, is said to be dominated by another chromosome, i_2, (for the present problem of minimization of $\boldsymbol{I}$ or maximization of $\boldsymbol{F}$), if

$$F_1^{(i_1)} < F_1^{(i_2)} \quad \text{(a)}$$

as well as

$$F_2^{(i_1)} < F_2^{(i_2)} \quad \text{(b)}$$

then

$$i_1 \text{ is dominated by } i_2 \quad \text{(c) (13)}$$

We test each of the N_P chromosomes in the population against all others to sort out *all* dominated chromosomes. As soon as a chromosome is found to be dominated, it is not checked for dominance with any other chromosome in the population. When all chromosomes have been checked for dominance, and all dominated chromosomes have been identified, the *rest of the chromosomes* are given a front number, *FRONT* = 1. These chromosomes having *FRONT* = 1 are called nondominated chromosomes.

3. All nondominated chromosomes are then assigned a *dummy* fitness value, F_1^*, equal to N_P.

4. Thereafter these *dummy* fitness values are modified according to the *sharing* procedure described in item 6 below, to assign a *shared fitness* value. Sharing is done to maintain diversity in the nondominated chromosomes.

5. In order to identify chromosomes for other fronts, we temporarily discard all nondominated chromosomes. The *remaining* chromosomes are again checked for dominance using Eq. (13) and new nondominated chromosomes are sorted and given a front number, *FRONT* = 2. Again, the new nondominated chromosomes (in *FRONT* 2) are given a dummy fitness value, F_2^*, which is slightly smaller than the lowest of the *shared* fitness values of the previous front. The sharing of the dummy fitness values is performed again, and a shared fitness value is assigned to each nondominated chromosome. This procedure is continued until all N_P chromosomes have been given a front number.

6. *Sharing*: Sharing is performed among the members of the i^{th} front (having n_i members) using the following procedure:

(a) For each chromosome, j, in front i, the dimensionless distance, d_{jk}, of this chromosome from any other chromosome, k, (including j) in the (same) front is calculated using

$$d_{jk} = \left\{ \sum_{i=1}^{N_{ga}} [(u_{P,i}^{(j)} - u_{P,i}^{(k)})/(u_{P,i}^{\max} - u_{P,i}^{\min})]^2 \right\}^{1/2} \quad (14)$$

(b) We calculate the *niche count*, m_j, using

$$m_j = \sum_{k=1}^{n_i} Sh(d_{jk}) \quad (15)$$

where

$$Sh(d_{jk}) = \begin{cases} 1 - \left(\dfrac{d_{jk}}{\sigma_{share}} \right)^{\alpha}, & \text{if } d_{jk} < \sigma_{share} \\ 0, & \text{otherwise} \end{cases} \quad (16)$$

σ_{share} is given by

$$\sigma_{share} = \frac{1}{2q^{1/(N_{ga}+1)}} \quad (17)$$

In Eq. (17), q is the number of Pareto optimal points desired (we have used q = 15 in our study). The parameter, α, is an exponent which controls the sharing effect (we have used α = 2 in the present study).

(c) The dummy fitness, F_i^*, of each chromosome, j, in front i, is modified by dividing F_i^* by the chromosome's niche count, m_j, to calculate the shared fitness value, $F_{ij}^{*\prime}$, as follows:

$$F_{ij}^{*\prime} = \frac{F_i^*}{m_j} \quad (18)$$

7. The stochastic remainder roulette wheel selection procedure is used on the shared fitness values, and a mating pool of N_P chromosomes is generated. This procedure involves proportionate selection, where first the number of copies made of each chromosome is equal to the integer part of the value of $F_{ij}^{*\prime}/\overline{F}_{ij}^{*\prime}$. Here, $\overline{F}_{ij}^{*\prime}$ is the average of the shared fitness values of all the N_P chromosomes in the population. Additional copies of the j^{th} chromosome in the i^{th} front (to make a total of N_P in the mating pool) are made thereafter, using a roulette wheel with probability proportional to the fractional part of $F_{ij}^{*\prime}/\overline{F}_{ij}^{*\prime}$.

8. After the mating pool is created, crossover takes place to produce the new population (next generation). This operation takes place at the chromosome (binary) level. Two chromosomes are randomly selected from the mating pool, a crossing site is selected (randomly again), and portions of the chromosomes before and after the crossing site are exchanged. For example, for seven-bit chromosomes with crossing site after the third binary, the crossover is described by the following:

$$\begin{array}{lll} 1\,0\,0 \,|\, 1\,1\,1\,1 & \longrightarrow & 1\,0\,0 \quad 0\,1\,0\,0 \\ 1\,1\,0 \,|\, 0\,1\,0\,0 & & 1\,1\,0 \quad 1\,1\,1\,1 \\ \text{(old generation)} & & \text{(new generation)} \end{array} \quad (19)$$

While performing crossovers, only $p_c N_P$ chromosomes are crossed, the remaining being left untouched (p_c is referred to as the crossover probability).

9. Another operation, called mutation, is also used to improve the next generation. The mutation operator

changes a binary number from 1 to 0 or vice versa, with a probability p_m. This operation is carried out for each of the bits in the population, again using appropriate random numbers. The need for mutation leads to a local search around the current solution and helps maintain the diversity of the population.

This completes one generation of NSGA. These sets of operations are carried out from one generation to the next until the number of generations equals the maximum number, Maxgen, specified at the starting of the program as an input parameter.

Acknowledgement: Financial support from the *Department of Science and Technology, Government of India, New Delhi* [through Grant III-5(59)/95-ET] is gratefully acknowledged. The correspondence author also gratefully acknowledges the hospitality and kindnesses (and the excellent infrastructure) that he is enjoying as a Visiting Professor at the *National University of Singapore*, where this paper was revised.

[1] J. R. Martin, J. F. Johnson, A. R. Cooper, *J. Macromol. Sci., Rev. Macromol. Chem.* **C8**, 57 (1972)
[2] R. W. Nunes, J. R. Martin, J. F. Johnson, *Polym. Eng. Sci*. **22**, 205 (1982)
[3] J. N. Farber, in *"Handbook of Polymer Science and Technology"*, N. P. Cheremisinoff, Ed., Marcel Dekker, New York 1989, Vol. 1, p. 429
[4] S. S. S. Chakravarthy, D. N. Saraf, S. K. Gupta, *J. Appl. Polym. Sci*. **65**, 529 (1997)
[5] K. Mitra, K. Deb, S. K. Gupta, *J. Appl. Polym. Sci*. **69**, 69 (1998)
[6] M. E. Sacks, S. I. Lee, J. A. Biesenberger, *Chem. Eng. Sci*. **28**, 241 (1973)
[7] E. Trommsdorff, H. Kohle, P. Lagally, *Makromol. Chem*. **1**, 169 (1947)
[8] K. Deb, *"Optimization for Engineering Design: Algorithms and Examples"*, Prentice Hall of India, New Delhi 1995
[9] D. E. Goldberg, *"Genetic Algorithms in Search, Optimization and Machine Learning"*, Addison-Wesley, Reading, MA 1989
[10] N. Srinivas, K. Deb, *Evolutionary Computation* **2**, 3 (1995)
[11] V. Chankong, Y. Y. Haimes, *"Multiobjective Decision Making – Theory and Methodology"*, Elsevier, New York 1983
[12] S. Garg, S. K. Gupta, D. N. Saraf, *J. Appl. Polym. Sci*., in press
[13] J. S. Vrentas, J. L. Duda, *AIChE J*. **25**, 1 (1979)
[14] V. Seth, S. K. Gupta, *J. Polym. Eng*. **15**, 283 (1995)
[15] R. B. B. Ram, S. K. Gupta, D. N. Saraf, *J. Appl. Polym. Sci.* **64**, 1861 (1997)
[16] S. K. Gupta, *"Numerical Methods for Engineers"*, New Age International, New Delhi 1995
[17] N. R. Vaid, S. K. Gupta, *Polym. Eng. Sci*. **31**, 1708 (1991)
[18] B. M. Louie, D. S. Soong, *J. Appl. Polym. Sci*. **30**, 3707 (1985)
[19] J. N. Farber, *Polym. Eng. Sci*. **26**, 499 (1986)
[20] A. Goicoechea, D. R. Hansen, L. Duckstein, *"Multiobjective Decision Analysis with Engineering & Business Applications"*, Wiley, New York 1982