

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/231372604>

# Dynamic Optimization of Batch Emulsion Polymerization Using MSIMPSA, a Simulated-Annealing-Based Algorithm

ARTICLE *in* INDUSTRIAL & ENGINEERING CHEMISTRY RESEARCH · OCTOBER 2004

Impact Factor: 2.59 · DOI: 10.1021/ie0496784

---

CITATIONS

12

---

READS

37

4 AUTHORS, INCLUDING:



**Gregory Francois**

University of Applied Sciences and Arts West...

61 PUBLICATIONS 299 CITATIONS

SEE PROFILE



**Romualdo L. Salcedo**

University of Porto

49 PUBLICATIONS 797 CITATIONS

SEE PROFILE

# Dynamic Optimization of Batch Emulsion Polymerization Using MSIMPISA, a Simulated-Annealing-Based Algorithm

R. M. Lima,<sup>†</sup> G. François,<sup>‡</sup> B. Srinivasan,<sup>‡</sup> and R. L. Salcedo<sup>\*,†</sup>

*Departamento de Engenharia Química, Faculdade de Engenharia da Universidade do Porto, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal, and Laboratoire d'Automatique, École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland*

The aim of this paper is to present an approach to dynamic off-line optimization of batch emulsion polymerization reactors using a stochastic optimizer. The control objective is to find the optimal temperature profile that minimizes the final batch time constrained by the final conversion and molecular weight. In this study, we evaluate the applicability of MSIMPISA, a simulated-annealing-based algorithm, to solve the optimal control problem. Two cases are studied: first, a simple case without energy balances and, second, a more realistic case using energy balances and constraints on heat transfer. In addition, an SQP optimizer was applied to perform a local optimization around the best results obtained by MSIMPISA. The following conclusions can be drawn from the results: (i) MSIMPISA can be applied in an easy and straightforward manner (black-box approach) to such optimal control problems. (ii) Even though MSIMPISA is a stochastic algorithm, the best obtained solution is so good that it cannot be further improved by local optimization methods.

## 1. Introduction

In dynamic optimization, the aim is to optimize a performance index (economic, quality, or safety indexes) by manipulating a time-varying process variable. For chemical batch reactors, the dynamic optimization optimizes a performance index through the manipulation of a process variable, such as a temperature profile, a feed rate addition profile (in the case of semibatch reactors), or the initial batch conditions (for example, the initial reactant concentrations).

The models can be based on only mass balances, or they can incorporate more detailed information, such as energy balances and constraints relative to the cooling system. Some authors<sup>1,2</sup> have used simple first-principles models to obtain an idea of the shape of batch reactor temperature profiles and theoretical maximum values<sup>3</sup> for performance indexes before employing more detailed models. The results obtained are not intended to be directly applied in a real batch reactor, but rather to provide useful information about the shape and tendency of process variable profiles.<sup>4</sup>

The dynamic optimization of batch polymerization reactor has mainly performed using gradient-based methods, such as Pontryagin's maximum principle, Lagrange multipliers, generalized reduced gradients (GRGs),<sup>5</sup> and sequential quadratic programming (SQP).<sup>1,6</sup> These techniques are prone to become stuck in local minima, and so, a considerable amount of effort is needed to choose an appropriate initial guess if one aims at attaining the best possible solution.

Global optimization methods are an alternative to local optimization methods whenever it is important and relevant to achieve the global optimum. Two different approaches can be employed: deterministic and sto-

chastic. Deterministic algorithms converge to a final solution for a given prespecified tolerance,<sup>7</sup> whereas the solutions obtained by stochastic algorithms are based on statistical methods involving probabilities. Although deterministic methods can guarantee global optimality for a specific type of problems, it should be noted that, for several general problems, no global optimization algorithm can guarantee the global optimum in a finite time. Therefore, the final solution should be considered only as the best obtained solution.

Recently, Papamichail and Adjiman<sup>7</sup> and Esposito and Floudas<sup>8</sup> proposed two effective methods based on the branch-and-bound solution of dynamic optimization problems. These algorithms are able to achieve the global optimum within a prespecified tolerance whenever certain requirements on the derivatives are fulfilled. Therefore, these algorithms are not appropriate for use with "black-box" simulation approaches, and this decreases the flexibility of application to general process engineering problems. This flexibility and the derivative free approach are some of the reasons that make stochastic optimizers a valuable alternative for the solution of general global optimization problems.

Stochastic algorithms can be broadly divided into four categories:<sup>9</sup> adaptive stochastic methods, evolutionary computation, clustering methods, and simulated annealing (SA). With stochastic optimizers, the global optimality of the final solution cannot be guaranteed; nevertheless, several studies have demonstrated the ability of such algorithms to reach the best known solutions. These algorithms employ strategies to escape local optima and ideally to reach the global optimum independently of the starting guess. Currently, stochastic or hybrid algorithms have a wide applicability, not only for solving nonlinear programming (NLP) and mixed-integer nonlinear programming (MINLP) problems but also for solving dynamic optimization problems. Examples include the optimizations of chemical batch reactors,<sup>10–14</sup> bioreactors,<sup>9,12,13,15</sup> and batch distillation.<sup>16,17</sup> In recent studies, SA algorithms were used

\* To whom correspondence should be addressed. E-mail: rsalcedo@fe.up.pt.

<sup>†</sup> Faculdade de Engenharia da Universidade do Porto.

<sup>‡</sup> École Polytechnique Fédérale de Lausanne.

in the dynamic optimization of a batch cooling crystallization;<sup>18</sup> of batch, semibatch, and nonisothermal evaporative crystallizations;<sup>19</sup> and of semibatch reactive crystallization processes.<sup>20</sup> Sarkar and Modak<sup>21</sup> presented a hybrid strategy combining artificial neural networks (ANNs) and an SA algorithm to obtain optimal control profiles. These authors advocate the choice of SA because of its capabilities of getting near global optimum solutions (escaping from local optima) associated with the simplicity of implementation.

In this study, more than trying to solve standard dynamic benchmark problems,<sup>14</sup> the main objective is to determine the applicability of an SA-based algorithm, MSIMPISA,<sup>22–24</sup> to the off-line dynamic optimization of the batch emulsion copolymerization of styrene and  $\alpha$ -methylstyrene. The purpose is to determine temperature profiles that minimize the final batch time subject to constraints on the quantity and quality of the final product and on the cooling system.

The optimization was performed using (i) the MSIMPISA optimizer, (ii) the adaptive random search algorithm MSGA,<sup>25,26</sup> and (iii) an SQP-based optimizer using the results from MSIMPISA as starting points.

In all cases, the problem was solved using a sequential approach in which the control function was discretized using proportional-length piecewise-constant, variable-length piecewise-constant, and variable-length piecewise-linear functions.

Recently, ANNs were used to generate continuous control profiles for dynamic optimization problems,<sup>21</sup> with the weights and bias for specific ANN models being the decision variables. This approach was able to achieve successful results for several dynamic optimization benchmarks. A different methodology was used by Choong and Smith,<sup>19</sup> who developed a profile generator able to generate control functions with the purpose of generating only practical and realizable profiles. In our study, the adopted parametrizations are based on a previous study<sup>27</sup> concerning the optimum shape of the control function for the present problem and on their simplicity of implementation. Additionally, the inclusion of a constraint in the derivative of the reactor temperature profile avoids abrupt reactor temperature changes. It is possible that, for other problems, ANN models or profile generators might prove to be more adequate.

Section 2 describes the SA-based algorithm and its implementation in the MSIMPISA optimizer. The problem formulation is presented in section 3. Section 4 reports the results obtained with MSIMPISA and an SQP optimizer. In this section, two cases are considered: the minimization of the final batch time using a simple model with the reactor temperature as the control variable and a second case including energy balances using the cooling fluid inlet temperature as the control variable. The MSIMPISA convergence properties are also evaluated through comparisons with MSGA, an SQP-based algorithm, and with published results.

The best solutions for these problems were obtained with the MSIMPISA optimizer, with final batch times about 1% better (lower) than published results. Also, MSIMPISA demonstrated the ability to obtain solutions near the best known optimum without any need to provide feasible initial guesses.

## 2. MSIMPISA Optimizer

The MSIMPISA optimizer was developed by Cardoso et al.<sup>22–24</sup> It is based on the combination of an SA

algorithm, the Metropolis algorithm,<sup>28–30</sup> and the nonlinear simplex method of Nelder and Mead.<sup>31</sup> This approach was first proposed by Press and Teukolsky<sup>32</sup> and was studied and improved by Cardoso et al.<sup>22–24</sup> for the efficient solution of combinatorial optimization problems,<sup>22</sup> nonconvex constrained NLP problems,<sup>23</sup> and MINLP problems.<sup>24,33</sup>

The Metropolis algorithm is based on an analogy with the physical process of annealing, whereby a solid is melted by increasing its temperature and then slowly cooled to allow crystallization into a state of minimum free energy.<sup>34</sup> The cooling rate and the initial temperature play an important role in the attainment of thermal equilibrium at each temperature to avoid the occurrence of locally optimal lattice structures, which translate into lattice imperfections. Thermal equilibrium at a given temperature is characterized by a Boltzmann distribution of the energy states. Under these conditions, a transition from a low to a high energy level can occur, and such transitions are assumed to be responsible for the system reaching a minimum-energy state. The algorithm starts with a high-energy state,  $E_n^j$ , and a corresponding high initial temperature value,  $T_{sa}^j$ . At this temperature level, a new energy state,  $E_{n+1}^j$ , is stochastically generated. The new energy state is accepted if  $E_{n+1}^j \leq E_n^j$ ; otherwise, it will be accepted as an improved state with a probability given by

$$P(\Delta E_{sa}) = \exp\left(\frac{E_{n+1}^j - E_n^j}{K_b T_{sa}^j}\right) \quad (1)$$

where  $T_{sa}^j$  is the current temperature level and  $K_b$  is Boltzmann's constant. When thermal equilibrium is reached, the temperature is decreased, and the probability of accepting a higher-energy state as an improved solution also decreases. By analogy with the physical process, in the Metropolis algorithm, the values of the objective function correspond to the energy states, and the minimum-energy state corresponds to the minimum of the optimization problem. The possible transitions to high-energy states correspond to "wrong-way" movements in optimization algorithms, employed to overcome local optima.<sup>25</sup> From an optimization point of view, the Metropolis algorithm is used as an acceptance/rejection probability criterion applied to the values of the objective function that becomes more stringent with the progress of the optimization procedure. Obviously, SA is not physical annealing, but this analogy is widely employed as a vehicle for understanding these algorithms. For the effective application of SA as an optimization algorithm, the main aspects that should be considered are (i) the specification of a high initial temperature value, which initially will allow all moves to be accepted; (ii) the adoption of a suitable cooling schedule that will avoid convergence to poor local optima; and (iii) the generation of new states (the continuous vector) at each temperature level.

In the MSIMPISA optimizer, the initial temperature is automatically generated according to a proposal by Aarst and Korst,<sup>34</sup> and the cooling schedule is the Aarst and van Laarhoven probabilistic schedule.<sup>35</sup> The nonlinear simplex method of Nelder and Mead<sup>31</sup> is applied to generate continuous system configurations, as suggested by Press and Teukolsky.<sup>32</sup> The vector of  $N$  decision variables is replaced by an  $(N + 1)$ -dimensional

simplex. During the simplex movements, each time a new vertex is generated, the simplex is perturbed in the function value of each vertex using a random variable and the corresponding value of the temperature level. The acceptance/rejection probability criterion from the Metropolis algorithm is then used to accept or reject the perturbed function value of the newly generated vertex. A detailed description of the initial simplex generation and approaches to handle the items above can be found elsewhere.<sup>14,22,23</sup> Thus, in the MSIMPISA algorithm, the role of SA in the overall approach is to allow for wrong-way movements, simultaneously providing (asymptotic) convergence to the global optimum, whereas the role of the nonlinear simplex is to generate new system configurations.

Effective handling of constraints is an important improvement over the original proposal by Press and Teukolsky.<sup>32</sup> Two approaches to deal with nonlinear constraints are implemented in MSIMPISA. The first assigns a very large positive value to the objective function it is being maximized, or a very large negative value otherwise, whenever an inequality constraint is violated or whenever the mathematical model cannot be solved. The second approach uses a penalizing scheme in which the maximum absolute violations of the inequality constraints associated with penalty coefficients are incorporated into the objective function. The penalty coefficients are problem dependent, depending on the objective function value and the violation of the constraints. Therefore, this option requires some tuning to properly define penalty coefficients. Details on the progression of the simplex algorithm and the Metropolis algorithm in the presence of penalized objective functions can be found elsewhere.<sup>23,24</sup> By specifying a null value for the temperature control parameter, MSIMPISA performs similarly to the simplex method of Nelder and Mead.<sup>31</sup> Thus, MSIMPISA is a substantially different SA-based algorithm than those used by Sarkar and Modak<sup>21</sup> and Choong and Smith.<sup>19</sup>

This algorithm showed good robustness, viz., good insensitivity to the starting point, and reliability in attaining the global optimum for a number of difficult NLP and MINLP problems described in the literature.<sup>23,24</sup> MSIMPISA was also applied to the synthesis of a nonequilibrium reactive distillation column, viz., a large-scale nonconvex problem, giving solutions very close to the global optimum.<sup>33</sup> Recently, it was compared with a random search method<sup>25</sup> for the solution of dynamic optimization benchmark problems, for which it demonstrated the ability to obtain solutions in the vicinity of the best known solution.<sup>14</sup> A comparison between MSIMPISA and two evolutionary algorithms (a genetic algorithm and an evolution strategy algorithm) to optimize MINLP problems<sup>36–38</sup> demonstrated that evolutionary algorithms did not improve over MSIMPISA in reaching the best known solution. Indeed, for one of the case studies presented, the MSIMPISA optimizer was the only one able to converge to the global optimum.

Generally, stochastic algorithms, and in this case MSIMPISA, do not need any a priori knowledge about the problem structure because the problem is treated as a black box. Therefore, MSIMPISA can be coupled with sophisticated and in-house simulation codes without requiring any preprocessing.

The CPU time required to perform an optimization run with MSIMPISA depends on the number of objective function evaluations (related to the number of inner

loops at each temperature level) and the evaluation time of each simulation step. Problems with finding feasible solutions might also contribute to increases in the CPU time required. Thus, the CPU time required is inherently related to the optimization problem to be solved. In this work, MSIMPISA is applied not to a benchmark problem, but rather to a real dynamic optimization problem with process constraints such as polymer properties at the final batch time and the reactor temperature derivative.

### 3. Problem Statement

**3.1. Mathematical Model.** The aim of this work is to evaluate the applicability of MSIMPISA, an SA-based optimizer, to the dynamic off-line optimization of the batch emulsion copolymerization of styrene and  $\alpha$ -methylstyrene. The model used in this study was developed and experimentally validated by Gentric et al.<sup>1</sup> and also used for a case study on a run-to-run dynamic optimization approach by François et al.<sup>27</sup> The model was developed with the purpose of being precise enough to describe the main phenomena involved in the process (in contrast to a detailed model), but also being able to be used to optimize and control a reactor. Thus, it can be considered a tendency model,<sup>39</sup> built with simplifications in the kinetic mechanism (modeling it as a homopolymerization reaction) and in the molecular weight distribution. Despite the simplifications, this model was able to predict conversions and molecular weights in agreement with experimental results, demonstrating that it was accurate enough to represent the relationship between the operating conditions, kinetics, and final product properties.<sup>1</sup> The mathematical equations that describe the model are presented in Appendix A.

To better understand the results obtained, it seems useful to briefly present the interpretation of the mechanisms involved. During an emulsion polymerization performed in a batch reactor, the relation between the reaction rate and time is divided into three different intervals. The first interpretation of this mechanism is due to Smith and Ewart.<sup>40</sup>

In interval I, free radicals are generated in the aqueous phase by initiator decomposition, and diffusion into micelles begins. The polymerization rate begins to increase.

In interval II, polymer begins to be formed inside polymer particles. The diffusion of the monomer into the particles is faster than the monomer consumption, favored by the high specific surface area of the particles. The particles are saturated with monomer. The disappearance of monomer droplets defines the end of this interval. The polymerization rate is almost constant.

In interval III, the polymerization continues using the monomer present in the particles. Because of the diminished monomer supply, the polymerization rate decreases with time.

**3.2. Formulation of the Optimization Problem.** The objective of this batch emulsion polymerization reactor problem is to minimize the final batch time,  $t_f$ , for a specified quality and quantity of polymer. The product quality is expressed through the specification of a minimum value of the number-average molecular weight,  $\bar{M}_{nf}$ , and the product quantity through the specification of a minimum conversion,  $X_f$ .

In a first approach, case I, the optimization was performed using a model with only the adopted kinetic



mechanism and the molecular weight distribution. Therefore, because of the lack of information relative to the reactor, the profiles obtained are intended not to be directly implemented,<sup>4</sup> but rather to provide information about the shape of the optimum profiles as well as an idea about the theoretical minimum performance index that can be expected.<sup>3</sup> The reactor temperature profile,  $T(t)$ , was chosen as the control variable, following previous work by Gentric et al.<sup>1</sup> It was bounded by a lower limit to guarantee a minimum reaction rate and by an upper limit due to operational constraints.

The optimization problem can be stated as

$$\min_{T(t), t_f} F_{\text{obj}} = t_f \quad (2)$$

subject to

Rate of monomer consumption

$$-\frac{dM}{dt} = R_p, \quad M(0) = M_0 \quad (3)$$

Rate of particle formation

$$\frac{dN_p}{dt} = \frac{R_i N_a}{1 + \frac{\epsilon N_p}{S}}, \quad N_p(0) = 0 \quad (4)$$

Zeroth-order moment of the MWD

$$\frac{dQ_0}{dt} = R_t + R_{\text{trM}}, \quad Q_0(0) = 0 \quad (5)$$

First-order moment of the MWD

$$\frac{dQ_1}{dt} = R_p, \quad Q_1(0) = 0 \quad (6)$$

$$X(t_f) \geq X_f \quad (7)$$

$$\bar{M}_n(t_f) \geq \bar{M}_{\text{nf}} \quad (8)$$

$$T_{\min} \leq T(t) \leq T_{\max} \quad (9)$$

where  $t_f$  is the final batch time,  $M$  is the global monomer concentration,  $R_p$  is the polymerization rate,  $N_p$  is the total number of particles,  $R_i$  is the initiator decomposition rate,  $\epsilon$  is the capturing efficiency of particles by micelles,  $S$  is the emulsifier concentration,  $Q_0$  is the zeroth-order moment of the molecular weight distribution,  $Q_1$  is the first-order moment of the molecular weight distribution,  $R_t$  is the termination rate,  $R_{\text{trM}}$  is the transfer-to-monomer rate,  $X$  is the polymerization conversion,  $X_f$  is the final polymerization conversion desired,  $\bar{M}_n$  is the number-average molecular weight, and  $\bar{M}_{\text{nf}}$  is the final number-average molecular weight desired. Note that omitting  $N_a$  from the original eq 11 in the paper of Gentric et al.<sup>1</sup> (see eq 4) is consistent with the value and units of  $\epsilon$  given by Gentric.<sup>41</sup>

In case II, the optimization was performed in a more realistic way considering additionally the energy balances, eqs 10 and 11, and a constraint on the reactor temperature derivative, eq 12.

$$\frac{dT_j}{dt} = \frac{F_j(T_{\text{jin}} - T_j)}{V_j} - \frac{UA(T_j - T)}{\rho_j V_j C_j}, \quad T_j(0) = T_{j0} \quad (10)$$

$$\frac{dT}{dt} = -\frac{V\Delta H}{m_r C_p} R_p + \frac{UA(T_j - T)}{m_r C_p}, \quad T(0) = T_0 \quad (11)$$

$$\left| \frac{dT}{dt} \right| \geq R_{\text{cmax}} \quad (12)$$

$$T_{\text{jin}_{\min}} \leq T_{\text{jin}}(t) \leq T_{\text{jin}_{\max}} \quad (13)$$

where  $T_j$  is the jacket temperature,  $T_{\text{jin}}$  is the cooling fluid flow rate,  $F_j$  is the cooling fluid temperature, and  $R_{\text{cmax}}$  is the reactor maximum cooling rate. Here, the control variable is the cooling fluid inlet temperature, instead of the reactor temperature. Thus, the problem formulation was changed to

$$\min_{T_{\text{jin}}(t), t_f} F_{\text{obj}} = t_f \quad (14)$$

subject to eqs 3–13.

Note that the inlet flow rate of the jacket fluid is constant and that its temperature is now the manipulated variable. A more practical case would be to consider the flow rates of cooling and heating fluids as manipulated variables, such as presented by Luus and Okongwu,<sup>3</sup> given that it is not realistic to perform abrupt changes in the temperature of the jacket inlet fluid. However, to obtain results that can be compared with those of Gentric et al.,<sup>1</sup> the jacket inlet fluid temperature was kept as the manipulated variable.

In addition, for case II, to minimize the polydispersity index,  $I_p$ , at the final batch time, a different objective function was considered

$$\min_{T_{\text{jin}}(t), t_f} F_{\text{obj}} = I_p(t_f) \quad (15)$$

subject to an additional equality constraint

$$\frac{dQ_2}{dt} = 2L^2(R_t + R_{\text{trM}}), \quad Q_2(0) = 0 \quad (16)$$

### 3.3. Parametrization of the Control Function.

The control variable or control function was parametrized using three different approaches: proportional-length piecewise-constant, variable-length piecewise-constant, and variable-length piecewise-linear.

**Proportional-Length Piecewise-Constant Control Policy.** For  $t \in [t_{\text{sw}}^k, t_{\text{sw}}^{k+1}]$ , the value of the control function is constant

$$T(t) = T^k \quad (17)$$

and

$$t_{\text{sw}}^{k+1} - t_{\text{sw}}^k = \frac{t_f}{P} \quad (18)$$

where  $P$  is the number of equal-length time intervals and  $t_{\text{sw}}$  represents the switching times that define the intervals. Therefore, the control function is defined by the specification of  $T^k$ ,  $t_f$ , and  $P$ .

**Variable-Length Piecewise-Constant Control Policy.** For  $t \in [t_{\text{sw}}^k, t_{\text{sw}}^{k+1}]$ , the value of the control function is constant

$$T(t) = T^k \quad (19)$$

However, the length of the time intervals is not constant. Thus, the control function is defined by  $T^k$ ,  $P$ ,  $t_{sw}^k$ , and  $t_f$ .

**Variable-Length Piecewise-Linear Control Policy.** For  $t \in [t_{sw}^k, t_{sw}^{k+1}]$ , the value of the control function is

$$T(t) = T^k + \left( \frac{T^{k+1} - T^k}{t_{sw}^{k+1} - t_{sw}^k} \right) (t - t_{sw}^k) \quad (20)$$

where the length of the time intervals for a specific final time is not constant. Thus, the control function is also defined by  $T^k$ ,  $P$ ,  $t_{sw}^k$ , and  $t_f$ .

## 4. Optimization Results

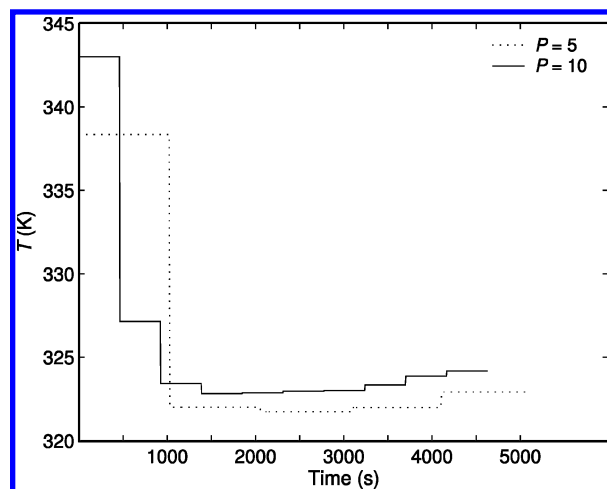
**4.1. Optimization Results Obtained with MSIMPSA.** The dynamic off-line optimization of the batch emulsion polymerization reactor presented in this work was first performed by Gentric et al.<sup>1</sup> to obtain optimal temperature profiles for an experimental reactor. The dynamic optimization problem was transformed into an NLP problem, through the use of orthogonal collocation, and then solved with an SQP-based method.

In this work, the approach is based on the parametrization of the control function, integration of the state equations, and use of the MSIMPSPA optimizer to update the vector of decision variables. The final state constraints as well as the constraint on the reactor temperature derivative were addressed using the strategies implemented in MSIMPSPA as described in section 2. In all cases, MSIMPSPA could easily find feasible points starting from infeasible ones; thus, the dynamic penalizing scheme described in section 2 was not necessary.

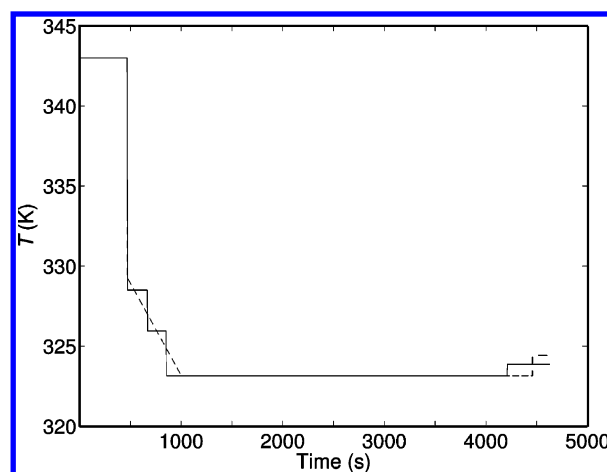
It is our aim to evaluate the applicability of the MSIMPSPA optimizer to these type of problems while searching for the best solution for this particular problem. Because MSIMPSPA and MSGA are stochastic-based optimizers, their performance must be evaluated on a statistical basis. Whenever the optimization was performed with either of these optimizers, 30 different seeds were used, which means that 30 optimization runs were performed using different randomly generated starting simplexes and points for MSIMPSPA and MSGA. The evaluation of the accuracy of the stochastic optimizers is thus based on the results of these 30 runs.

The simulation with the model of the batch emulsion polymerization used by MSIMPSPA and MSGA were coded in Fortran 77, whereas the MatLab language was used with the SQP-based algorithm. The integration of the ordinary differential equation (ODE) system in case I was performed using a Runge–Kutta (4,5) pair method implemented in the package *rksuite* obtained from the digital library netlib (www.netlib.org), with tolerances set to  $10^{-10}$ , whereas in case II, the ODE system was solved using the software package DASPK<sup>42</sup> with tolerances set to  $10^{-6}$ , as this is a stiff problem.

**4.1.1. Case I.** As a first approach, the process was optimized without considering the energy balances and using the reactor temperature as the control variable. Initially, the control variable was parametrized using a piecewise-constant control policy with different values of  $P$ . Because different values of  $P$  can give different optimization results,<sup>10</sup> solutions obtained with  $P = 1$ , 5, and 10 are presented. With  $P = 1$ , the best isothermal reactor operation obtained was  $\{T^{iso}, t_f\} = \{327.6 \text{ K}, 7141.5 \text{ s}\}$ . Figure 1 shows the optimal reactor temper-



**Figure 1.** Reactor temperature profiles discretized with proportional-length piecewise-constant control policies.



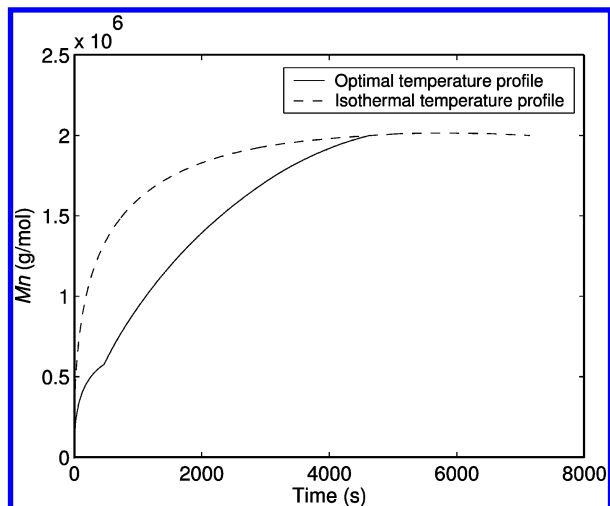
**Figure 2.** Optimal reactor temperature profiles for case I: (—) variable-length piecewise-constant control with  $P = 5$ , (---) variable-length piecewise-constant control with  $P = 4$  using piecewise-linear control for  $t \in [t_{sw}^1, t_{sw}^2]$ .

ature profiles obtained with  $P = 5$  and  $P = 10$ , which correspond to the final batch times  $t_f = 5127.8 \text{ s}$  and  $t_f = 4630.4 \text{ s}$ , respectively. The improvement obtained upon increasing the number of time intervals might suggest that a higher value of  $P$  would improve the performance index; however better results were not obtained with  $P = 50$ .

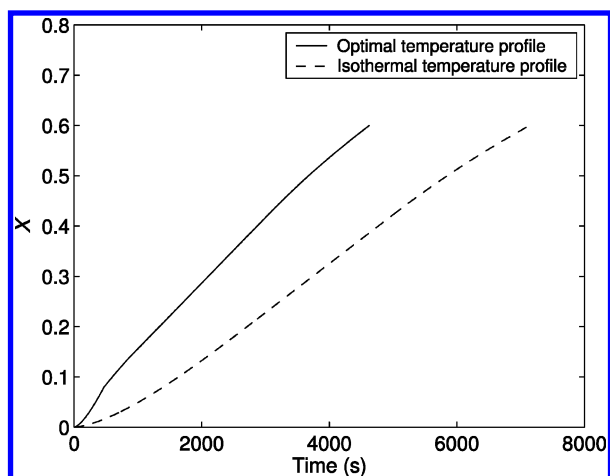
This problem was also solved using a variable-length piecewise-constant control policy with  $P = 5$ . Here, the vector of decision variables is  $[t_f, t_{sw}^j, T^k]$ , with  $j = 1, \dots, 4$  and  $k = 1, \dots, 5$ . The minimum final batch time achieved was  $t_f = 4628.9 \text{ s}$  with the reactor temperature profile shown in Figure 2.

There is a slight improvement in the performance index and a reactor temperature profile similar to the one obtained with a piecewise-constant control policy with  $P = 10$ . Figures 3 and 4 show the number-average molecular weight and conversion profiles obtained with this last result and with the corresponding isothermal profile. In these figures, the improvements that can be achieved with an optimal reactor temperature profile are obvious as the same quality and quantity of product can be obtained within 64.8% of the final batch time for an isothermal operation.

To obtain an improved performance index, the second and third time intervals of the reactor temperature



**Figure 3.** Number-average molecular weight profiles obtained with an isothermal profile and with an optimal temperature profile. The final batch times are 7141.5 and 4628.9 s, respectively.

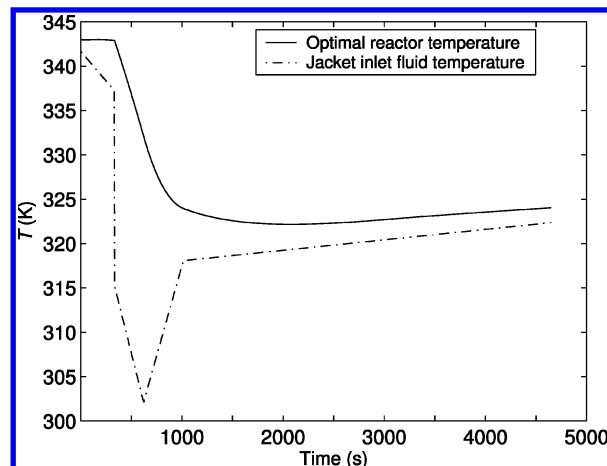


**Figure 4.** Polymerization conversion profile obtained with an isothermal profile and with an optimal temperature profile. The final batch times are 7141.5 and 4628.9 s, respectively.

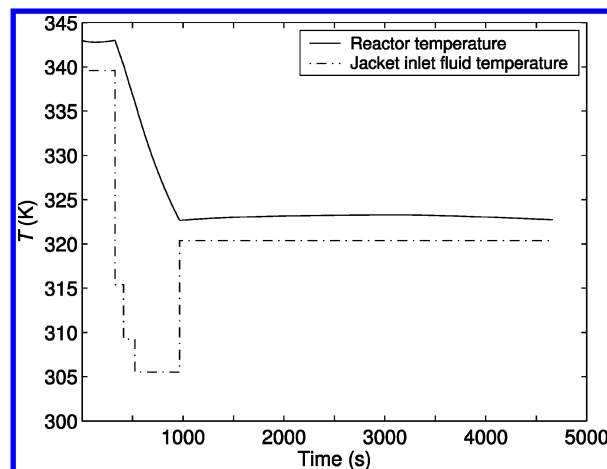
profile were replaced by piecewise-linear control, with the first time interval set equal to the upper bound. However, the final batch time obtained decreased only to  $t_f = 4628.8$  s. Both reactor temperature profiles are presented in Figure 2.

As already mentioned, Gentric et al.<sup>1</sup> converted the system of differential equations into an algebraic system of equations using orthogonal collocation on finite elements, transforming this dynamic optimization problem into an NLP problem. With this approach the final batch time is obtained from interpolation using the knots from the solution of the NLP problem. Their minimum final batch time was  $t_f = 4595$  s, which is better than the best solution obtained with MSIMPSA. However, by interpolation, at the final time, the terminal constraints have the values  $\bar{M}_{nf} = 1.98 \times 10^6$  g mol<sup>-1</sup> and  $X_f = 0.5934$ , instead of  $\bar{M}_{nf} = 2.0 \times 10^6$  g mol<sup>-1</sup> and  $X_f = 0.60$ . Thus, to evaluate the best final batch time obtained from MSIMPSA, another case was considered using a variable-length piecewise-constant control policy with  $P = 5$  and relaxing the constraints to  $\{\bar{M}_{nf}, X_f\} = \{1.98 \times 10^6$  g mol<sup>-1</sup>, 0.5934 $\}$ , achieving a best final batch time of  $t_f = 4538.3$  s.

Qualitatively, the optimal reactor temperature profiles obtained have the same tendency as those presented by Gentric et al.,<sup>1</sup> with the exception of the



**Figure 5.** Optimal temperature profiles for case II obtained using a variable-length piecewise-linear parametrization with  $P = 5$ .



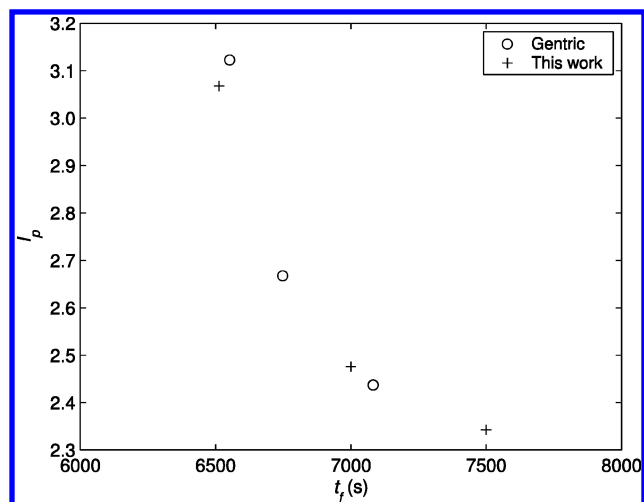
**Figure 6.** Optimal temperature profiles for case II obtained using a variable-length piecewise-constant parametrization with  $P = 5$ .

decrease in temperature after the first time interval. The optimal reactor temperature profile presented by these authors has an abrupt decrease from the maximum temperature to the minimum temperature, whereas in this study, the best solution shows some steps before the temperature reaches the minimum value.

**4.1.2. Case II: Minimization of the Final Batch Time.** In this case, the energy balances were included in the mathematical model, and the process was optimized considering a constraint on the maximum reactor cooling rate. Thus, a more realistic case that is theoretically more difficult to solve is presented.

Here, the cooling fluid inlet temperature is defined as the control variable and the reactor temperature profile as a state variable. The shapes of the reactor temperature profiles from case I were not used as starting points or even as constraints on the cooling fluid inlet temperature profiles. However, those results were used to select as a first approach a variable-length piecewise-constant parametrization with  $P = 5$  and, additionally, a variable-length piecewise-linear discretization with  $P = 5$ .

The minimum final batch time obtained was  $t_f = 4649.9$  s, given by the optimal cooling fluid inlet temperature profile presented in Figure 5, using the piecewise-linear parametrization. Figure 6 shows the optimal temperature profiles that correspond to the minimum final batch time  $t_f = 4663.1$  s, obtained with the variable-length piecewise-constant parametrization.



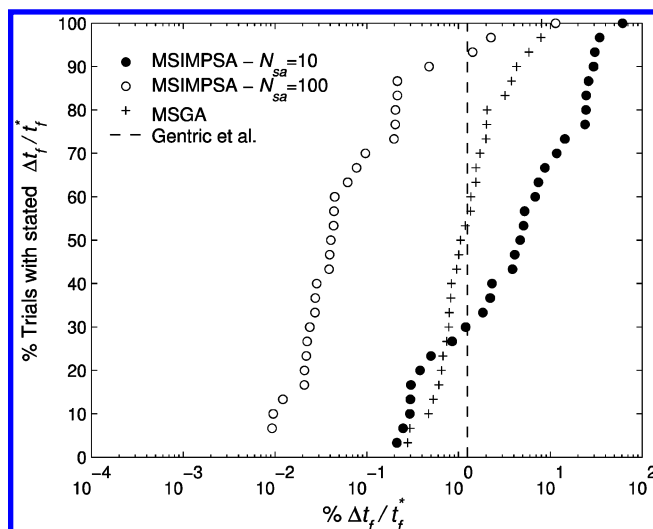
**Figure 7.** Optimum polydispersity index for different maximum values of  $t_f$  obtained using a variable-length piecewise-linear parametrization with  $P = 5$ .

Qualitatively, both jacket inlet fluid temperature profiles gave reactor temperature profiles with shapes similar to those presented by Gentric et al.<sup>1</sup>

**4.1.3. Case II: Minimization of the Final Batch Time and Polydispersity Index.** The optimization of the final batch time and the polydispersity index of the final polymer involves the minimization of two performance indexes. Although MSIMPISA was not developed with the purpose of solving multicriteria optimization problems, several alternatives are available, usually involving penalty functions.<sup>11,43</sup>

Gentric<sup>41</sup> solved this problem by changing the objective function to  $F_{\text{obj}} = \gamma_1 t_f + \gamma_2 I_p(t_f)$  and performing a study on the influence of  $\gamma_1$  and  $\gamma_2$  on the final batch time and polydispersity index. However, such an approach involves a study of the best values of the weighting factors to use. In this work, the use of penalty or weight factors was avoided, and the polydispersity index was minimized with maximum allowable values for the final batch times. First, the final batch time was minimized for  $\bar{M}_{\text{nf}} = 3 \times 10^6$  g/mol and  $X_f = 0.6$  because these were the values used by Gentric,<sup>41</sup> giving  $t_f = 6513.3$  s and  $I_p = 3.176$ . Three more optimizations were performed minimizing the polydispersity index, constrained by  $t_f \leq \{6513.3, 7000, 7500\}$  s. Figure 7 shows the optimized polydispersity index and how it is affected by the final batch time. The results follow the general trend obtained by Gentric.<sup>41</sup>

**4.1.4. Evaluation of the MSIMPISA Results.** The results presented above correspond to the optimal solutions that have been found. However, for each case, 30 optimization runs using different initial points were made. In this section, the distributions of the results obtained with MSIMPISA are compared with the results of Gentric et al.<sup>1</sup> and with the results obtained with MSGA, an adaptive random search algorithm. Both algorithms were used with standard option values. In the MSIMPISA algorithm, the parameter  $N_{\text{sa}}$  defines the number of inner loops at each temperature (parameter of the algorithm) and, consequently, the number of function evaluations. It is one of the parameters that has direct influence on the computational times, as well as on the accuracy of the solutions obtained. Thus, for case I, results with different values of  $N_{\text{sa}}$  are presented to provide an idea about how this parameter can influence the optimization results. For each distribution,



**Figure 8.** Cumulative results from MSIMPISA and MSGA for case I obtained using a variable-length piecewise-constant parametrization with  $P = 5$ ,  $X_f = 0.5934$ ,  $\bar{M}_{\text{nf}} = 1.98 \times 10^6$  g/mol, and  $t_f^* = 4538.3$  s.

the number of function evaluations and typical average execution times for a Pentium 4 computer at 1.7 GHz are presented.

Figure 8 shows the distribution of the final batch times from case I obtained with the parametrization that led to the best results, i.e., a variable-length piecewise-constant parametrization. These results correspond to the optimizations performed with  $X_f = 0.5934$  and  $\bar{M}_{\text{nf}} = 1.98 \times 10^6$  g/mol (see section 4.1.1). Although the results show some dependency on the initial solution vector (starting simplex), they also show that, for  $N_{\text{sa}} = 100$ , 93.3% of the optimization runs gave better results than the result presented by Gentric et al.,<sup>1</sup> with the minimum final batch time being 1.23% lower. For this case, using the MSIMPISA algorithm with only the nonlinear simplex activated, only 2 of 30 optimization runs obtained a lower final batch time than reported previously.<sup>1</sup> Compared to the MSGA algorithm, the MSIMPISA optimizer presented a superior accuracy, with 86.7% of the results giving a higher accuracy when used with  $N_{\text{sa}} = 100$ . The average numbers of objective function evaluations for each distribution with the MSIMPISA and MSGA algorithms are presented in Table 1.

The average execution time with  $N_{\text{sa}} = 100$  was 31.1 min. For  $N_{\text{sa}} = 10$ , the results presented a lower accuracy, but still with 30% of the optimization runs giving a better result, and the average execution time decreased to 2.6 min. With the MSGA algorithm, the average execution time was 9.6 min.

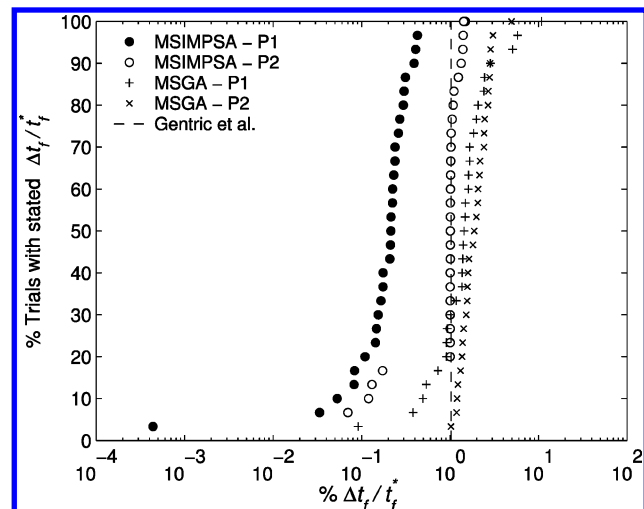
For case II, Figure 9 shows the distribution of the results obtained with the MSIMPISA and MSGA algorithms for the two parametrizations used. For each parametrization, apart from the constraints  $X_f = 0.6$  and  $\bar{M}_{\text{nf}} = 2.0 \times 10^6$  g/mol, the optimizations were also performed with the constraints  $X_f = 0.5924$  and  $\bar{M}_{\text{nf}} = 2.03 \times 10^6$  g/mol, corresponding to the final batch time presented by Gentric et al.,<sup>1</sup> and  $N_{\text{sa}} = 100$ . First, from Figure 9, it is possible to identify that the parametrizations used gave different results with the MSIMPISA algorithm, with the variable-length piecewise-constant parametrization giving a higher percentage of better results, which is somewhat surprising. Comparing these results with those obtained by Gentric et al.,<sup>1</sup> the



**Table 1. Average Number of Function Evaluations for Each Case Using Different Parameterizations<sup>a</sup>**

	MSIMPSPA			MSGGA
	$N_{sa}$	$T_{sa}^0$	$N_{fobj}$	$N_{fobj}$
Case I				
$P_1$	10	$0.52 \times 10^{13}$	47016	198171
$P_1$	100	$0.44 \times 10^{13}$	567512	
Case II				
$P_1$	10	$0.65 \times 10^4$	28438	125581
$P_1$	100	$0.57 \times 10^4$	356518	
$P_2$	10	$0.85 \times 10^4$	32792	131472
$P_2$	100	$0.49 \times 10^4$	436320	

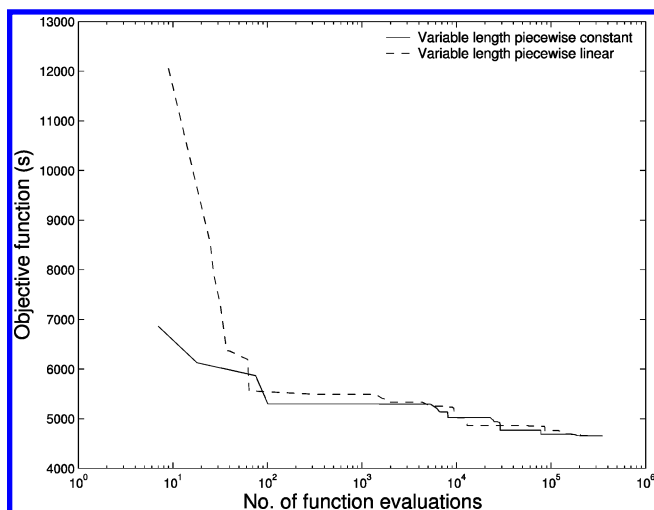
<sup>a</sup>  $P_1$  = variable-length piecewise-constant parameterization.  $P_2$  = variable-length piecewise-linear parameterization.



**Figure 9.** Cumulative results from MSIMPSPA and MSGGA for case II, obtained with  $X_f = 0.5924$ ,  $\bar{M}_{nf} = 2.03 \times 10^6$  g/mol, and  $t_f^* = 4652.4$  s.  $P_1$ , variable-length piecewise-constant parameterization;  $P_2$ , variable-length piecewise-linear parameterization.

MSIMPSPA optimizer was able to improve the final batch time by 1.01%, revealing a high efficiency (96.7%) in obtaining values lower than the previously published result. For a variable-length piecewise-linear parameterization with  $N_{sa} = 100$ , the results are in accordance with those of Gentric et al.,<sup>1</sup> with 16.7% of optimization runs giving better results. For this case, MSIMPSPA also revealed greater accuracy than the MSGGA algorithm, independently of the type of parametrization used. The respective average numbers of function evaluations for each distribution are presented in Table 1. Typical convergence curves for the best final batch times obtained with these parametrizations, shown in Figure 10, reveal that, after approximately the first 100 function evaluations, the two parametrizations presented similar trajectories.

The average execution times for MSIMPSPA with a variable-length piecewise-constant parametrization were 2.4 and 30.0 min, whereas with a variable-length piecewise-linear parametrization, they were 1.9 and 25.6 min, respectively, for  $N_{sa} = 10$  and  $N_{sa} = 100$ . With MSGGA, the average execution times were 11.3 and 8.0 min for the variable-length piecewise-constant and linear parametrization, respectively. The MSIMPSPA algorithm with only the nonlinear simplex activated was also applied to this problem using the variable-length piecewise-constant parametrization, but in 30 optimization runs, no final batch times lower than the one presented by Gentric et al.<sup>1</sup> were obtained.



**Figure 10.** Typical trajectories for MSIMPSPA for case II with two different parametrizations.

**4.2. Optimization Using an SQP-Based Algorithm.** Stochastic algorithms, when applied to optimization problems, are able to reach the global optimum asymptotically, independently of the starting guess. Nevertheless, because of the high number of function evaluations, it might be practical only to reach the vicinity of the global optimum. In contrast, SQP methods show a much faster convergence provided a starting guess close enough to the optimum is provided. Thus, it is our purpose to evaluate whether providing the best results from MSIMPSPA as starting points to the SQP-based optimizer yields better results. A similar strategy was reported by Carrasco and Banga.<sup>44</sup> These authors developed two hybrid algorithms based on stochastic and deterministic algorithms, where the stochastic algorithms were responsible for providing solutions in the vicinity of the global optimum that could be further used as starting points for the deterministic algorithms.

It is not our aim to perform an overall comparison between the MSIMPSPA and the SQP-based optimizer, but rather to evaluate the quality of the solutions obtained from MSIMPSPA. The simulation code was written in MatLab language, and for both cases, the ODE system was solved with the MatLab function *ode45*, based on an explicit Runge–Kutta (4,5) formula; for the optimization, the MatLab function *fmincon* was used.

For each case, one optimization was performed starting from the best result from MSIMPSPA, and several optimizations were then performed with these initial points perturbed by  $\pm 1\%$  and  $\pm 5\%$ . For case I, the reactor control variable was discretized using a variable-length piecewise-constant control policy with  $P = 5$ , whereas for case II, the jacket inlet fluid temperature was discretized using a variable-length piecewise-linear control policy with  $P = 5$ . Table 2 reports the final batch times obtained for each case.

First, for both cases, there was no improvement of the minimum final batch time, either using the best starting point from MSIMPSPA or using the perturbed starting profiles. For perturbations of the starting profile of 1%, in both cases, the SQP optimizer did not converge to the best solution, but the final batch times were not too far from the optimum value. However, for perturbations of 5%, the final batch times presented a significant error relative to the best solution.

**Table 2. Minimum Final Batch Times Obtained by SQP with Different Starting Points**

starting profile	case I <sup>a</sup>		case II <sup>b</sup>	
	$t_f$ (s)	$\epsilon_r^c$ (%)	$t_f$ (s)	$\epsilon_r^c$ (%)
$\theta_0$	4629.0	0	4649.9	0
0.99 $\theta_0$	4643.3	0.31	4667.5	0.38
1.01 $\theta_0$	4641.9	0.28	4666.2	0.35
0.95 $\theta_0$	7000.0	51.2	5041.0	8.41
1.05 $\theta_0$	5570.1	20.3	4686.9	0.80

<sup>a</sup> Case I:  $\theta_0 = [469.52, 668.45, 854.33, 4210.80, 343.00, 328.50, 325.94, 323.14, 323.86]$ . <sup>b</sup> Case II:  $\theta_0 = [333.17, 333.23, 621.82, 1014.53, 341.67, 337.24, 315.28, 302.11, 318.08, 322.37]$ . <sup>c</sup>  $\epsilon_r$  = error relative to  $\theta_0$ .

## 5. Conclusions

An SA-based optimizer, MSIMPSPA, was applied to find the minimum final batch time of a batch emulsion polymerization reactor both without and with energy balances and constraints on the cooling rate. It was applied as an alternative to local optimization methods, which are prone to become stuck in local minima. MSIMPSPA was able to determine optimum reactor temperature profiles that minimize final batch times, with constraints on the state variables and on the reactor cooling rate. MSIMPSPA was not used with any specific initial point or heuristics related to the kinetic mechanism. Improved final batch times were obtained for both case studies when compared to those of Gentric et al.,<sup>1</sup> with optimum reactor temperature profiles presenting the same tendency. MSIMPSPA was also capable of predicting optimal polydispersity indexes subject to the same constraints in agreement with the general trend observed in the literature.<sup>41</sup> MSIMPSPA was also more accurate, based on a comparison of 30 optimization runs with standard option values, than an available adaptive random search method,<sup>25,26</sup> being able to arrive at lower reactor final batch times. On average, CPU times were on the order of 30 min with a Pentium 4 based computer, with a high robustness (93–96%) in improving previously reported solutions. These execution times can be dramatically reduced (down to about 2–3 min), but with a penalty on the fraction of better results obtained (30%).

An SQP-based algorithm was also applied to test the quality of the solutions obtained by MSIMPSPA, but improved solutions could not be obtained. The SQP-based algorithm is also quite sensitive to the starting point. This suggests that a hybrid implementation such as MSIMPSPA (stochastic SA plus deterministic nonlinear simplex of Nelder and Mead<sup>31</sup>) is quite successful, as a local algorithm cannot improve its solutions.

This work shows that the MSIMPSPA optimizer has the capability of solving off-line dynamic optimization problems, and as such, it can be a valuable tool for the optimization of batch polymerization reactors.

## Acknowledgment

This work was partially supported by FCT (Portuguese Foundation for Science and Technology) under Contract PRAXIS XXI/BD/21481/99 and by European Union RTN Project BatchPro HPRN-CT-2000-00039, employing the computational facilities of Instituto de Sistemas e Robótica (ISR).

## Appendix A

The mathematical model for the batch emulsion copolymerization reactor of styrene and  $\alpha$ -methylstyrene is presented in this appendix. It is a tendency model, based on assumptions and simplifications that consider the following kinetic steps: radical formation, particle nucleation, radical capture by micelles and particles, termination propagation, and transfer to monomer. All details on the model derivation can be found elsewhere.<sup>1</sup> The rates of monomer consumption and particle formation and the zeroth-, first-, and second-order moments of the molecular weight distribution are given in eqs 3–6 and 16, respectively. The model of the kinetic mechanism is presented with first the equations of the rate constants and then the respective reaction rate equations.

Rate of initiator decomposition

$$k_d = k_{d0} e^{-E_{kd}/RT} \quad (\text{A.1})$$

$$R_i = 2f_i k_d I_0 \quad (\text{A.2})$$

where  $I_0$  is the initiator concentration, assumed constant;  $f_i$  is the efficiency factor of initiator decomposition; and  $R_i$  is the rate of initiator decomposition.

Rate of transfer to monomer

$$k_{\text{styp}} = e^{bf_{\text{ms}}} \quad (\text{A.3})$$

$$k_{\text{trM}0} = k_{\text{styp}} k'_{\text{trM}0} \quad (\text{A.4})$$

$$k_{\text{trM}} = k_{\text{trM}0} e^{-E_{\text{ktrM}}/RT} \quad (\text{A.5})$$

$$R_{\text{trM}} = k_{\text{trM}} M_p \frac{N_p \bar{n}}{N_a} \quad (\text{A.6})$$

where  $f_{\text{ms}}$  is the initial  $\alpha$ -methylstyrene molar fraction,  $\bar{n}$  is the average number of radicals per particle,  $M_p$  is concentration of monomer in the particles,  $N_p$  is the number of particles,  $N_a$  is Avogadro's number, and  $R_{\text{trM}}$  the rate of transfer to monomer.

Rate of propagation

$$k_{\text{styp}} = e^{af_{\text{ms}}} \quad (\text{A.7})$$

$$k_{p0} = k_{\text{styp}} k'_{p0} \quad (\text{A.8})$$

$$k_p = k_{p0} e^{-E_{kp}/RT} \quad (\text{A.9})$$

$$R_p = k_p \frac{N_p}{N_a} M_p \bar{n} \quad (\text{A.10})$$

where  $R_p$  is the rate of propagation and the monomer concentration is constant during intervals I and II and is calculated from

$$X(t) = 1 - \frac{M(t)}{M_0} \quad (\text{A.11})$$

$$X_c = \frac{M_{pc} M_M - \rho_m}{\left(1 - \frac{\rho_m}{\rho_p}\right) M_{pc} M_M - \rho_m} \quad (\text{A.12})$$

$$M_p = \begin{cases} M_{pc} & \text{if } X \leq X_c \\ ((1-x)\rho_m/[(1-X) + X(\rho_m/\rho_p)]M_M & \text{if } X > X_c \end{cases} \quad (\text{A.13})$$

Rate of particle formation

$$k_v = \left( \frac{36\pi M_M^2}{\omega_p^2 \rho_p^2 (a_s N_a)^3} \right)^{1/3} \quad (\text{A.14})$$

$$S = S_0 - k_v (X M_0)^{2/3} N_p^{1/3} \quad (\text{A.15})$$

$$R_t = \frac{R_i \bar{n}}{1 + \frac{S}{N_p \epsilon}} \quad (\text{A.16})$$

where  $S_0$  is the initial emulsifier concentration and  $\bar{n}$  is the average number of radicals per particle.

Kinetic chain length

$$L = \frac{R_p}{R_t + R_{trM}} \quad (\text{A.17})$$

The conversion, number- and weight-average molecular weights, and polydispersity index are calculated at the final time from the integration of the system of equations 3–6, as follows

$$X(t_f) = 1 - \frac{M(t_f)}{M_0} \quad (\text{A.18})$$

$$\bar{M}_n(t_f) = M_M \frac{Q_1(t_f)}{Q_0(t_f)} \quad (\text{A.19})$$

$$\bar{M}_w(t_f) = M_M \frac{Q_2(t_f)}{Q_1(t_f)} \quad (\text{A.20})$$

$$I_p(t_f) = \frac{\bar{M}_w(t_f)}{\bar{M}_n(t_f)} \quad (\text{A.21})$$

Table 3 lists all data related to this problem.

**Table 3. Model Parameters**

parameter	value	units	parameter	value	units
$a_s$	$1.8 \times 10^5$	$\text{dm}^2 \text{g}^{-1}$	$\bar{n}$	0.5	
$a$	-7.3		$R_{cmax}$	-0.04	$\text{K s}^{-1}$
$b$	-4		$\rho_j V_j C_j$	1.946	$\text{J K}^{-1}$
$E_{kd}$	140.2	$\text{kJ mol}^{-1}$	$\rho_m$	0.91	$\text{kg L}^{-1}$
$E_{kp}$	29	$\text{kJ mol}^{-1}$	$\rho_p$	1.1	$\text{kg L}^{-1}$
$E_{ktrM}$	85	$\text{kJ mol}^{-1}$	$S_0$	4.432	$\text{g L}^{-1}$
$\epsilon$	$5 \times 10^{-16}$	$\text{g part}^{-1}$	$T_0$	343	K
$f_i$	0.5		$T_{j0}$	343	K
$f_{ms}$	$8.96 \times 10^{-2}$		$T_{jinmax}$	343	K
$I_0$	$3.7 \times 10^{-3}$	$\text{mol L}^{-1}$	$T_{jinmin}$	293	K
$k_{d0}$	$4.5 \times 10^6$	$\text{s}^{-1}$	$T_{max}$	343	K
$k'_{p0}$	$5.7 \times 10^6$	$\text{L mol}^{-1} \text{s}^{-1}$	$T_{min}$	313	K
$k'_{trM0}$	$1.5 \times 10^{11}$	$\text{L mol}^{-1} \text{s}^{-1}$	$UA$	6.4	$\text{J K}^{-1} \text{s}^{-1}$
$M_0$	2.16	$\text{mol L}^{-1}$	$V\Delta H$	-66.9	$\text{kJ L mol}^{-1}$
$M_{pc}$	5.38	$\text{mol L}^{-1}$	$X_f$	0.6	
$m_r C_p$	4.151	$\text{kJ K}^{-1}$	$\omega_p$	0.4223	
$M_M$	105.41	$\text{g mol}^{-1}$			
$M_{nf}$	$2.0 \times 10^6$	$\text{g mol}^{-1}$			

## Nomenclature

$A$  = area ( $\text{m}^2$ )  
 $a$  = coefficient in eq 13

$a_s$  = area occupied by 1 g of emulsifier ( $\text{dm}^2$ )  
 $b$  = coefficient in eq 10  
 $C_j$  = cooling fluid heat capacity ( $\text{J kg}^{-1} \text{K}^{-1}$ )  
 $E_{kd}$  = activation energy, initiator decomposition ( $\text{kJ mol}^{-1}$ )  
 $E_{kp}$  = activation energy, propagation ( $\text{kJ mol}^{-1}$ )  
 $E_{ktrM}$  = activation energy, transfer to monomer ( $\text{kJ mol}^{-1}$ )  
 $E^j$  = simulated annealing parameter  
 $F_j$  = cooling fluid flow rate ( $\text{L s}^{-1}$ )  
 $f_i$  = initiator efficiency  
 $f_{ms}$  = initial  $\alpha$ -methylstyrene molar fraction  
 $I_0$  = initial initiator concentration ( $\text{mol L}^{-1}$ )  
 $I_p$  = polydispersity index  
 $K_b$  = Boltzmann constant  
 $k_d$  = rate constant of initiator decomposition ( $\text{s}^{-1}$ )  
 $k_{d0}$  = rate constant of initiator decomposition ( $\text{s}^{-1}$ )  
 $k_p$  = rate constant of propagation ( $\text{L mol}^{-1} \text{s}^{-1}$ )  
 $k_{p0}$  = rate constant of propagation ( $\text{L mol}^{-1} \text{s}^{-1}$ )  
 $k_{trM}$  = rate constant of transfer to monomer ( $\text{L mol}^{-1} \text{s}^{-1}$ )  
 $k_{trM0}$  = rate constant of transfer to monomer ( $\text{L mol}^{-1} \text{s}^{-1}$ )  
 $k_{styp}$  = styrene homopolymerization propagation rate constant  
 $k_{styptr}$  = styrene homopolymerization transfer rate constant  
 $k_v$  = kinetic constant of monomolecular layer adsorption  
 $M$  = global monomer concentration ( $\text{mol L}^{-1}$ )  
 $M_M$  = monomer molecular weight ( $\text{g mol}^{-1}$ )  
 $m_r C_p$  = reactor total heat capacity ( $\text{J K}^{-1}$ )  
 $\bar{M}_n$  = number-average molecular weight ( $\text{g mol}^{-1}$ )  
 $\bar{M}_{nf}$  = number-average molecular weight at the final batch time ( $\text{g mol}^{-1}$ )  
 $M_p$  = monomer concentration in the particles ( $\text{mol L}^{-1}$ )  
 $M_{pc}$  = critical monomer concentration in the particles ( $\text{mol L}^{-1}$ )  
 $\bar{M}_w$  = weight-average molecular weight ( $\text{g mol}^{-1}$ )  
 $\bar{n}$  = average number of radicals per particle  
 $N$  = number of inactive particles per volume unit (particle  $\text{L}^{-1}$ )  
 $N_a$  = Avogadro's number  
 $N_{fobj}$  = number of function evaluations  
 $N_p$  = total number of particles per unit volume (particle  $\text{L}^{-1}$ )  
 $n_s$  = aggregation number of micelles  
 $N_{sa}$  = number of inner loops, simulated annealing  
 $P$  = number of time intervals  
 $Q_i$  =  $i$ th moment of the molecular weight distribution  
 $R$  = universal gas constant ( $\text{J mol}^{-1} \text{K}^{-1}$ )  
 $R_{cmax}$  = maximum cooling rate of the reactor ( $\text{K s}^{-1}$ )  
 $R_i$  = initiator rate constant ( $\text{mol L}^{-1} \text{s}^{-1}$ )  
 $R_p$  = polymerization rate ( $\text{mol L}^{-1} \text{s}^{-1}$ )  
 $R_t$  = termination rate ( $\text{mol L}^{-1} \text{s}^{-1}$ )  
 $R_{trM}$  = rate of transfer to monomer ( $\text{mol L}^{-1} \text{s}^{-1}$ )  
 $S$  = emulsifier concentration ( $\text{g L}^{-1}$ )  
 $T$  = batch reactor temperature (K)  
 $T_j$  = jacket temperature (K)  
 $T_{jin}$  = jacket inlet fluid temperature (K)  
 $T_{sa}^j$  = simulated annealing cooling schedule parameter at outer loop  $j$   
 $t_f$  = final batch reactor time (s)  
 $t_{sw}$  = switching time (s)  
 $U$  = heat-transfer coefficient ( $\text{J K}^{-1} \text{m}^{-2} \text{s}^{-1}$ )  
 $V$  = reactor volume (L)  
 $V_j$  = reactor jacket volume (L)  
 $x_i$  = fraction  
 $X$  = polymerization conversion  
 $X_c$  = critical conversion  
 $X_f$  = conversion at the final batch time

## Greek Symbols

$\Delta H$  = polymerization reaction enthalpy ( $\text{J mol}^{-1}$ )  
 $\epsilon$  = capturing efficiency of particles by micelles ( $\text{g particle}^{-1}$ )  
 $\epsilon_r$  = relative error  
 $\theta$  = vector with reactor temperature profile parameters



$\rho_j$  = cooling fluid density (g L<sup>-1</sup>)

$\rho_m$  = monomer density (g L<sup>-1</sup>)

$\rho_p$  = polymer density (g L<sup>-1</sup>)

$\omega_p$  = polymer weight fraction in the particles

#### Subscripts

0 = initial conditions

#### Superscripts

\* = optimum

#### Acronyms

ANN = artificial neural networks

GRG = generalized reduced gradient

MINLP = mixed-integer nonlinear programming

MSGA = MINLP Salcedo-Gonçalves-Azevedo algorithm

MSIMPSA = MINLP simplex simulated annealing algorithm

MWD = molecular weight distribution

NLP = nonlinear programming

SA = simulated annealing

SQP = successive quadratic programming

#### Literature Cited

- (1) Gentric, C.; Pla, F.; Latifi, M. A.; Corriou, J. P. Optimization and nonlinear control of a batch emulsion polymerization reactor. *Chem. Eng. J.* **1999**, 75, 31.
- (2) Aziz, N.; Mujtaba, I. M. Optimal operation policies in batch reactors. *Chem. Eng. J.* **2002**, 85, 313.
- (3) Luus, R.; Okongwu, O. N. Towards practical optimal control of batch reactors. *Chem. Eng. J.* **1999**, 75, 1.
- (4) Bonvin, D. Optimal operation of batch reactors—A personal view. *J. Process Control* **1998**, 8, 355.
- (5) Jang, S. S.; Lin, P. H. Discontinuous minimum end-time temperature/initiator policies for batch emulsion polymerization of vinyl acetate. *Chem. Eng. Sci.* **1991**, 46, 3153.
- (6) Silva, D. C. M.; Oliveira, N. M. C. Optimization and nonlinear model predictive control of batch polymerization systems. *Comput. Chem. Eng.* **2002**, 26, 649.
- (7) Papamichail, I.; Adjiman, C. Global optimization of dynamic systems. *Comput. Chem. Eng.* **2004**, 28, 403.
- (8) Esposito, W.; Floudas, C. Deterministic global optimization in nonlinear optimal control problems. *J. Global Optim.* **2000**, 17, 97.
- (9) Banga, J. R.; Balsa-Canto, E.; Moles, C. G.; Alonso, A. A. Dynamic Optimization of bioreactors—A review. *Proc. Indian Natl. Sci. Acad.* **2003**, 69A, 257.
- (10) Luus, R. Optimal control of batch reactors by iterative dynamic programming. *J. Process Control* **1994**, 4, 218.
- (11) Bojkor, B.; Luus, R. Optimal control of nonlinear systems with unspecified final times. *Chem. Eng. Sci.* **1996**, 51, 905.
- (12) Banga, J. R.; Seider, W. D. Global optimization of chemical processes using stochastic algorithms. In *State of the Art in Global Optimization*, Floudas, C. A., Pardalos, P. M., Eds.; Kluwer Academic Publishers: Dordrecht, The Netherlands, 1996; Vol. 7, pp 563–583.
- (13) Carrasco, E. F.; Banga, J. R. Dynamic optimization of batch reactors using adaptive stochastic algorithms. *Ind. Eng. Chem. Res.* **1997**, 36, 2252.
- (14) Salcedo, R. L.; Lima, R. P.; Cardoso, M. F. Simulated annealing for the global optimization of chemical processes. *Proc. Indian Natl. Sci. Acad.* **2003**, 69A, 359.
- (15) Banga, J. R.; Alonso, A. A.; Singh, R. P. Stochastic dynamic optimization of batch and semicontinuous bioprocesses. *Biotechnol. Prog.* **1997**, 13, 326.
- (16) Hanke, M.; Li, P. Simulated annealing for the optimization of batch distillation processes. *Comput. Chem. Eng.* **2000**, 24, 1.
- (17) Li, P.; Löwe, K.; Arellano-Garcia, H.; Wozny, G. Integration of simulated annealing to a simulation tool for dynamic optimization of chemical processes. *Chem. Eng. Process.* **2000**, 39, 357.
- (18) Choong, K.; Smith, R. Optimization of batch cooling crystallization. *Chem. Eng. Sci.* **2004**, 59, 313.
- (19) Choong, K.; Smith, R. Novel strategies for optimization of batch, semi-batch and heating/cooling evaporative crystallization. *Chem. Eng. Sci.* **2004**, 59, 329.
- (20) Choong, K.; Smith, R. Optimization of semi-batch reactive crystallization processes. *Chem. Eng. Sci.* **2004**, 59, 1529.
- (21) Sarkar, D.; Modak, J. ANNSA: A hybrid artificial neural network/simulated annealing algorithm for optimal control problems. *Chem. Eng. Sci.* **2003**, 58, 3131.
- (22) Cardoso, M. F.; Salcedo, R. L.; Feyer de Azevedo, S. Nonequilibrium simulated annealing: A faster approach to combinatorial minimization. *Ind. Eng. Chem. Res.* **1994**, 33, 1908.
- (23) Cardoso, M. F.; Salcedo, R. L.; Feyer de Azevedo, S. The simplex-simulated annealing approach to continuous nonlinear optimization. *Comput. Chem. Eng.* **1996**, 20, 1065.
- (24) Cardoso, M. F.; Salcedo, R. L.; Feyer de Azevedo, S.; Barbosa, D. A simulated annealing approach to the solution of MINLP problems. *Comput. Chem. Eng.* **1997**, 21, 1349.
- (25) Salcedo, R.; Gonçalves, M. J.; Feyer de Azevedo, S. An improved random-search algorithm for nonlinear optimization. *Comput. Chem. Eng.* **1990**, 14, 1111.
- (26) Salcedo, R. L. Solving nonconvex nonlinear programming and mixed-integer nonlinear programming problems with adaptive random search. *Ind. Eng. Chem. Res.* **1992**, 31, 262.
- (27) François, G.; Srinivasan, B.; Bonvin, D. Run-to-Run Optimization of Batch Emulsion Polymerization. In *IFAC World Congress 2002*; Elsevier Science Ltd.: Amsterdam, The Netherlands, 2002; p 1258.
- (28) Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H.; Teller, E. Equation of state calculations by fast computing machines. *J. Chem. Phys.* **1953**, 21, 1087.
- (29) Kirkpatrick, S.; Gelatt, C. D.; Vecchi, M. P. Optimization by simulated annealing. *Science* **1983**, 220, 671.
- (30) Kirkpatrick, S. Optimization by simulated annealing: Quantitative studies. *J. Stat. Phys.* **1984**, 34, 975.
- (31) Nelder, J. A.; Mead, R. A simplex-method for function minimization. *Comput. J.* **1965**, 7, 308.
- (32) Press, W. H.; Teukolsky, S. A. Simulated annealing over continuous spaces. *Comput. Phys.* **1991**, 5, 426.
- (33) Cardoso, M. F.; Salcedo, R. L.; Feyer de Azevedo, S.; Barbosa, D. Optimization of reactive distillation processes with simulated annealing. *Chem. Eng. Sci.* **2000**, 55, 5059.
- (34) Aarts, E. H. L.; Korst, J. *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*; Wiley: New York, 1989.
- (35) Aarts, E. H. L.; van Laarhoven, P. J. M. Statistical cooling: A general approach to combinatorial optimization problems. *Philips J. Res.* **1985**, 40, 193.
- (36) Cardoso, M.; Salcedo, R.; de Azevedo, S.; Barbosa, D. Untitled. *Comput. Chem. Eng.* **2002**, 26, 141.
- (37) Costa, L.; Oliveira, P. Evolutionary algorithms approach to the solution of mixed integer nonlinear programming problems. *Comput. Chem. Eng.* **2001**, 25, 257.
- (38) Oliveira, P.; Costa, L. Untitled—Reply. *Comput. Chem. Eng.* **2002**, 26, 143.
- (39) Filippi, C.; Greffe, J. L.; Bordet, J.; Villiermaux, J.; Barnay, J. L.; Bonte, P.; Georgakis, C. Tendency modeling of semibatch reactors for optimization and control. *Chem. Eng. Sci.* **1986**, 41, 913.
- (40) Smith, W. V.; Ewart, R. H. Kinetics of emulsion polymerization. *J. Chem. Phys.* **1948**, 16, 592.
- (41) Gentric, C. Optimisation dynamique et commande non linéaire d'un réacteur de polymérisation en émulsion/par Caroline Gentric. Ph.D. Thesis, INPL, Nancy, France, 1997.
- (42) Brown, P. N.; Hindmarsh, A. C.; Petzold, L. R. Using Krylov methods in the solution of large-scale differential-algebraic systems. *SIAM J. Sci. Comput.* **1994**, 15, 1467.
- (43) Silva, C. M.; Biscaia, E. C. Genetic algorithm development for multi-objective optimization of batch free-radical polymerization reactors. *Comput. Chem. Eng.* **2003**, 27, 1329.
- (44) Carrasco, E.; Banga, J. A hybrid method for the optimal control of chemical processes. In *UKACC International Conference on Control '98*; 1998; Vols. I & II, pp 925–930.

Received for review April 21, 2004

Revised manuscript received July 27, 2004

Accepted August 30, 2004

IE0496784