Simulated Annealing For The Optimization Of Chemical Batch Production Processes[1]

Michael Hanke[2]
Royal Institute of Technology
Department of Numerical Analysis and Computer Science
S–10044 Stockholm
Sweden

and

Pu Li[3]
Institute of Process and Plant Technology
Technical University of Berlin
D–10623 Berlin
Germany

November 16, 1998

[1]Part of this work was done while the first author was working at the Weierstraß Institute of Applied Analysis and Stochastics Berlin, Germany

[2]Fax number: +46 (8) 790 0930. Email: hanke@nada.kth.se

[3]Fax number: +49 (30) 314 26915. Email: li@dynamik.fb10.tu-berlin.de

Abstract

Batch distillation processes are widely used in chemical industry. In this work, we consider the optimization of such processes by simulated annealing. Although this method is stochastically in nature, it has two evitable advantages: it can be readily connected to highly sophisticated simulation codes and it converges towards a global optimum. According to the characteristics of batch distillation operation we propose to use a two-step computation approach. A feasible strategy (admissible control) will be searched for in the first step and it will be optimized in the second step. The approach has been applied to three models of batch distillation ranging from a simple test example to a real production system. These results show the potential of the method for developing optimal operation strategies for batch chemical processes.
Keywords: batch distillation, simulated annealing, dynamic optimization.

1 Introduction

The determination of optimal control strategies for chemical processes with respect to a maximal gain in the products, minimal production time, and maximal profit is one of the main goals in the design and operation of production processes [19]. Distillation processes are among the most important technologies for the separation of multiple component mixtures. Distillation can be done either in a continuous operational regime (where the mixture is continuously supplied and the products are continuously led off) or in a discontinuous regime – the so-called batch distillation process –, where the apparatus is loaded with the mixture at startup time and the different fractions are taken out one by one. While the mathematical modeling of the former gives rise to a large system of nonlinear equations, the latter leads to a large system of coupled nonlinear differential and non-differential (algebraic) equations. Such a system is referred to as a *differential-algebraic* equation system. Thus, the numerical simulation of batch distillation processes is a challenging task which requires a lot of computing resources. On the other hand, the interest on the simulation and optimization of batch processes is growing since such processes have a number of advantages over continuous processes: they are useful for the simultaneous separation of different species with a single column, they are more flexible, and they are necessary for the production of very pure chemicals.

Because of the importance of batch distillation processes, a number of simulation packages have been developed. They are in daily use in research and development and have reached a rather advanced stage of sophistication, stability, and robustness. Consequently, they are huge (several hundreds of thousands of source lines) and contain a lot of technological know-how. For industrial applications it is important to implement optimization algorithms within the framework of these simulation packages. In principle, two solutions are possible:

1. While it is easy to compute the value of the cost functional, its derivatives are not easily available. So the first approach is to use the simulation package as a black box, and to apply a standard optimization technique (like SQP) with numerically approximated

gradients.

2. The second approach consists of a modification of the numerical kernel of the simulation package such that the derivatives of the numerical approximation with respect to the (discretized) controls are computed additionally. In the case of a multiple shooting approach this leads to the so-called internal differentiation while in the case of a full discretization approach (e.g. collocation methods) one is confined to compute the derivatives explicitly.

While the first solution is rather straightforward and easily implementable it has some serious drawbacks: (i) The computation of admissible controls, i.e. such controls that guarantee the fulfillment of *all* constraints (especially the purity and security constraints), can be an expensive and cumbersome problem. (ii) The use of numerically approximated gradients leads to convergence problems.

The second approach results in a fast converging algorithm if the initial guess is close to a (local) optimum. The problem of admissible controls remains. On the other hand, the implementation is very expensive and, for some chemical processes, even impossible without essential changes in the given simulation packages.

The third property is common to both approaches: due to the strong nonlinearity of the problem, only local convergence can be guaranteed and the obtained result represents a local optimum.

All these problems encountered in practical applications stimulated the search for alternatives for the optimization of chemical batch processes. In a number of papers, stochastic methods are proposed for the solution of certain optimization problems in chemical plant design (see e.g. [2, 5, 7, 8, 11, 12, 17]). The present paper is devoted to the study of a variant of the *simulated annealing* method applied to the problem at hand. Although simulated annealing was originally developed for discrete optimization problems, generalizations to continuous problems are known today. On the other hand, reports on dynamic optimization by using simulated annealing have not been found to our knowledge. Simulated annealing has a number of interesting features [22]:

- The algorithm uses only the values of the cost functional. Thus the simulation package can be used as a black box.

- The methods converges towards a global minimum (in probability).

On the other hand, due to the stochastic nature of the method the speed of convergence can be very slow.

The aim of this paper is to show the applicability and the limitations of the simulated annealing method applied to the optimal control of batch processes. It turns out that it has some promising properties when trying to determine admissible starting controls. After obtaining these controls the convergence slows down such that the coupling with a super-linearly convergent method should be considered.

In Section 2 we present the simulated annealing algorithm and its modification used. Examples for the performance of simulated annealing are given in Section 3. A simple example taken from [4] was implemented as an `Octave` [3] script. The second example is taken from [14] and is based on the simulation package `DOSOK` [13] which uses a piecewise polynomial collocation method for the simulation. The third example was communicated to the authors by B. HEGNER. For its treatment we use the package `CHEMASIM` developed by BASF AG, Ludwigshafen [6].

2 Batch Distillation Processes and Simulated Annealing

2.1 The Mathematical Model of Batch Distillation

The basic mathematical model for the description of a batch distillation process is a large (structured) system of differential-algebraic equations

$$f(\dot{x}, x, t, u) = 0 \tag{1}$$

where t belongs to the interval $t_0 \leq t \leq t_F$. Here, x is the vector of state variables (e.g., masses, concentrations, pressures, temperatures, etc.) while u denotes the vector of control variables (e.g. reflux ratio). The constraints on the state variables and the control variables are given by

$$x_{j,\min} \leq x_j(t) \leq x_{j,\max}, \qquad j \in J \tag{2}$$

$$u_{k,\min} \leq u_k(t) \leq u_{k,\max}, \qquad k \in K \tag{3}$$

as well as by conditions of the type

$$\int_{t_i}^{t_{i+1}} x_j(t)dt \geq a_j, \qquad j \in J \tag{4}$$

(mass constraints) or

$$\frac{\int_{t_i}^{t_{i+1}} x_j(t)x_l(t)dt}{\int_{t_i}^{t_{i+1}} x_j(t)dt} \geq b_j, \qquad j \in J, l \in L \tag{5}$$

(purity constraints) on certain subintervals $[t_i, t_{i+1}] \subseteq [t_0, t_F]$ which represent the main cuts (yield parts) of the whole production cycle. The switching points t_i as well as the end time t_F belong to the control parameters. The cost functional φ can be represented in the general form

$$\varphi = g_1(x(t_F), t_F) + \int_{t_0}^{t_F} g_2(x(t), u(t), t)\,dt. \tag{6}$$

Very often, the control u is discretized by a piecewise constant function where the switching points t_i are included in the set of points of discontinuity of the control. At this point, it does not make much sense to introduce very complicated expressions for the control since it must be realizable in the technical environment without too much effort.

Taking into account that (1) is a differential-algebraic equation system (usually with index 1), one is faced with discontinuities of the state vector x at the points of discontinuity of u. Therefore, at every breakpoint, the numerical solution of (1) can lead to the problem of a consistent initialization for the initial value solver.

2.2 Simulated Annealing

Simulated annealing is a stochastic optimization procedure based on the analogy with the annealing of solids [12, 22]. Assume that a solid body is heated such that it is completely melted. Every configuration of its particles should have the same probability to occur. Now the temperature is slowly lowered. For each temperature level, the state of the solid is assumed to reach thermal equilibrium. The probability of being in a state s_i with energy E_i is given by the Boltzmann distribution

$$w_T(s_i) = \frac{1}{Q(T)} \exp\left(\frac{E_i}{k_B T}\right)$$

where k_B denotes the Boltzmann constant and $Q(T)$ is a certain normalizing factor. If the temperature T decreases sufficiently slowly such that the equilibrium distribution is reached for each temperature, then the states with lower energy will be preferred. In the limit $T \to 0$, the solid is frozen and the limiting probability distribution is an equidistribution where the support is exactly the set of states with (global) minimal energy.

This heuristics gives rise to a *Monte-Carlo* method: Assume that we have to minimize a functional $\varphi : \to \mathbb{R}$ over the domain $D \subseteq \mathbb{R}^n$. Let us start with a fixed temperature T and a given state $x_0 \in D$. Now, a sequence $\{x_i\}$ of states is randomly generated. To given $x_i \in D$ a state $\bar{x}_{i+1} \in D$ is generated in a neighborhood of x_i. The new state $\bar{x}_{i+1}$ is accepted unconditionally if $\varphi(\bar{x}_{i+1}) \leq \varphi(x_i)$. Otherwise, the probability of accepting $\bar{x}_{i+1}$ as a new state x_{i+1} is given by $\exp\left(-\frac{\varphi(\bar{x}_{i+1})-\varphi(x_i)}{k_B T}\right)$ if $\varphi(\bar{x}_{i+1}) \geq \varphi(x_i)$. Having done a sufficiently large number of steps the temperature can be decreased. In contrast to random walk algorithms the energy is not strictly decreasing in this procedure. With some positive probability, increasing energy will be accepted thus allowing a certain "hill climbing".

Originally, simulated annealing was proposed for problems from combinatorial optimization [10, 24]. Today, a number of proposals for generalizing the idea to continuous global optimization are known (e.g. [1, 9, 15, 20, 21, 23]).

Algorithm 1 is a rather informal description of simulated annealing.

```
simulated_annealing(x_in, x_out, T_max, T_min, k_max)
x := x_in;
T := T_max;
while T ≥ T_min do
    for k = 1 to k_max do
        x' := generate(x,T);
        if φ(x') < φ(x) then x := x'
        elif rand < ψ(φ(x') − φ(x)) then x := x'
        endif
    enddo
    T := cool(T);
endwhile
x_out := x;
```

Algorithm 1: Sketch of a simulated annealing algorithm for minimizing $\varphi : D \to \mathbb{R}$

For a concrete realization, a number of ingredients must be discussed:

1. Acceptance criterion ($\texttt{rand} < \psi(\varphi(x') - \varphi(x))$).

`rand` is a random number distributed uniformly over $[0,1]$. As discussed above originally $\psi(c) = \exp(-\frac{c}{k_B T})$ was chosen [16] which is known as the *Metropolis criterium* today. But a number of other criteria can be used as well [22].

2. State generation (`generate`)

   For each $x \in D$, a neighborhood $D(x,T) \subseteq D$ is defined. Sometimes it is given naturally by the problem at hand (e.g., if D is an open set, some ball around x can be chosen). In other cases, a distance and/or connection is defined artificially (e.g. in combinatorial problems). According to some probability distribution (most often uniform) an element x' is drawn from $D(x,T)$ randomly.

3. Cooling schedule (`cool`(T), $T_{\min}$, $T_{\max}$)

   The choice of a correct cooling schedule is a rather hard problem. But it is crucial to the performance and convergence of the method. In order to obtain fast convergence, T should be decreased as fast as possible. On the other hand, if the cooling is to fast, then the algorithm may not reach the global minimum. According to the theory a cooling schedule proportional to $\log(n)^{-1}$ is necessary in order to guarantee global convergence. Here, n denotes the number of times T is decreased. Since this is very slow, a simple strategy $T := \alpha T$ with some $0 \ll \alpha < 1$ is used frequently. The choice of $T_{\max}$ and $T_{\min}$, respectively, may be critical, too. Sophisticated techniques for choosing the parameters are discussed in [22].

4. Length of the Markov chain ($k_{\max}$)

   In principle, this choice belongs to the cooling schedule. $k_{\max}$ should be large enough for a given temperature to approach the equilibrium distribution close. On the other hand, no computing time should be wasted. So $k_{\max}$ is usually small (even $k_{\max} = 1$ is possible).

2.3 Putting It Together

In case of a batch distillation process we are faced with two problems:

1. Computation of admissible controls. While it is straightforward to fulfill the simple constraints (2), (3), a control satisfying the other constraints are not easily available. Note that the latter ones can only be checked after a full simulation run. So their computation is quite expensive.

2. Optimization of (6).

Moreover, the amounts and purities depend very sensitive on the control parameters. The set of admissible controls is usually narrow in some directions of the parameter space. Therefore, the definition of the neighborhood $D(u,T)$ of an admissible control should be chosen in such a way that it reflects the topology. Since the latter is not known in advance, a certain adaptivity is necessary. Therefore, we decided to proceed in two steps: First, admissible initial values are computed by using an objective function

$$\Omega(u) = \sum_{m=1}^{M} \alpha_m \max(0, \beta_m - h_m(u)) \tag{7}$$

if a constraint has the form

$$h_m(u) \geq \beta_m.$$

Here, α_m is a positive scaling factor. The simple constraints (3) are not included in (7) but taken into account explicitly. So both problems are solved by essentially the same procedure with the only difference of some tuning parameters in the simulated annealing algorithm. The computation of admissible initial values is stopped if $\Omega(u) = 0$, and this control is put into the subsequent optimization of (6).

In order to use the gain of information obtained during the solution process we applied an approach due to VANDERBILT and LOUIE [23]. The definition of $D(u)$ is given by the following considerations: Assume that n is the dimension of D, that is $u \in \mathbb{R}^n$. Let $(w_1, \dots, w_n)^T$ denote a sample of n independent variables distributed uniformly on the interval $[-\sqrt{3}, \sqrt{3}]$ (i.e. with zero mean and unit variance). The stepsize δu for $\bar{u} = u + \delta u$ is chosen according to

$$\delta u = Qw$$

where Q controls D and, consequently, the distribution of the steps. This gives $D = Q([-\sqrt{3}, \sqrt{3}]^n)$. The step distribution can be characterized by the covariance matrix $\mathbf{s}$ which is given by $\mathbf{s} = QQ^T$. Reading this equation from the other side, one can generate random steps with any desired covariance matrix $\mathbf{s}$ by a Cholesky decomposition of $\mathbf{s}$. A choice of $\mathbf{s}$ is used which should match the Hessian of the objective function. This is done by computing the first and second moments of a random walk of $R \geq n$ steps $\{u^r\}$:

$$\begin{aligned} A_m &= \frac{1}{R} \sum_{r=1}^{R} u_m^r \\ S_{mk} &= \frac{1}{R} \sum_{r=1}^{R} (u_m^r - A_m)(u_k^r - A_k). \end{aligned}$$

Now, $\mathbf{s}$ is chosen according to

$$\mathbf{s} = \frac{\chi_s}{\gamma R} S. \tag{8}$$

A value for γ can be obtained from statistical considerations: its value is approximately 0.11. The "growth factor" $\chi_s > 1$ is recommended in [23] to be typically of the order of magnitude 3. We found out that this leads to a large number of rejected steps. So we have chosen a value slightly larger than 1. At the beginning of the annealing process, no information is available. Therefore, the first guess for Q is the identity matrix where the entries are scaled by the expected range of variation of the variables multiplied by a security factor of 0.01. According to the proposal in [23], the length R is chosen as $15n$.

The stopping criterium for the optimization process is given by

$$\langle \varphi \rangle - \varphi_{\min} \leq \eta |\langle \varphi \rangle|$$

where $\langle \varphi \rangle$ denotes the mean value after the last R samples while $\varphi_{\min}$ is the minimum value reached so far.

3 Examples

3.1 A Model by S. FARHAT *et al.*

The batch distillation process is modeled using the following assumptions:

- theoretical plates,
- negligible liquid holdup,
- negligible pressure drop,
- total condenser,
- constant vapor and liquid flow,
- no feed stream and intermediate sidestream,
- thermodynamic constants depending only on the temperature.

This leads to the set of equations given in Fig. 1 [4].

In the special model, $n = 3$ components and $m - 2 = 5$ plates were used. The values of D and V are taken as constants. The initial concentrations X_{11} and X_{21} as well as the initial value

Overall mass balance:

$$\frac{dB}{dt} = \frac{-V}{R+1} = -D$$

Mass balance of the $n-1$ lightest components between condenser and reboiler:

$$B\frac{dX_{i1}}{dt} = \frac{-V}{R+1}(X_{im} - X_{i1}), \qquad i = 1,\ldots,n-1$$

Summation equation for the reboiler:

$$\sum_{i=1}^{n} X_{i1} = 1$$

equilibrium equation for the reboiler:

$$\sum_{i=1}^{n} K_{i1}(T_1)X_{i1} = 1$$

Partial mass balance of the $n-1$ lightest components between a plate p and the condenser:

$$K_{ip}(T_p)X_{ip} - \frac{R}{R+1}X_{i,p+1}\frac{1}{R+1}X_{im} = 0, \qquad i = 1,\ldots,n-1, \quad p = 2,\ldots,m-1$$

Equilibrium equation on vapor compositions at plate p:

$$\sum_{i=1}^{n} K_{ip}(T_p)X_{ip} = 1, \qquad p = 2,\ldots,m-1$$

Summation equation for plate p:

$$\sum_{i=1}^{n} X_{ip} = 1, \qquad p = 2,\ldots,m-1$$

Summation equation for the condenser:

$$\sum_{i=1}^{n} X_{im} = 1$$

The following notations are used above:

m	number of plates including reboiler and condenser
n	number of components
B	holdup in the reboiler
X_{ip}	concentration of component i on plate p
D	distillate flow rate
V	vapor flow rate
R	reflux ratio
K_{ip}	vapor/liquid equilibrium coefficient
T_p	temperature on plate p

Figure 1: Batch process model used in [4]

for B are given. The initial data for the other unknowns must be computed from the algebraic part of the equations in Fig. 1.

The equilibrium coefficients are approximated by the Antoine model which leads to expressions

$$\ln K_{ip}(T_p) = \ln K_i(T_p) = A_i - \frac{B_i}{T_p + C_i}.$$

The control variables in the present problem are the (time dependent) reflux ratio and the switching point between production cuts and off-cuts.

Since there are components to be separated, the objective function is simply to maximize the gain

$$P_1 + P_2$$

subject to purity constraints

$$g_k \geq b_k, \qquad k = 1, 2$$

on both components. This leads to three switching points $t_0 = 0 < t_1 < t_2 < t_3$. It holds

$$P_j = \int_{t_{2j-2}}^{t_{2j-1}} \frac{V}{R(t)+1} dt, \qquad j = 1, 2$$

and

$$g_k = \frac{1}{P_k} \int_{t_{2k-1}}^{t_{2k-2}} \frac{V X_{km}(t)}{R(t)+1} dt, \qquad k = 1, 2.$$

On every subinterval, $R(t)$ is taken to be constant. Typical simulation results are recorded in Figures 2 and 3.While the first two figures show the history for computing consistent initial values, the latter ones show the optimization history. Instead of using the values obtained by the consistent initialization phase we used the “optimal” control variables of [4] as a starting guess for the optimization run. Our results are given in Table 1. The value of the objective function for that values is 62.888. These computations were done in the `Octave` system.

t_1	t_2	R_1	R_2	R_3
1.239	1.337	2.978	7.493	2.701

Table 1: Optimal control parameters

3.2 Batch Distillation of a Ternary System

The basic simulation model of this example is a more detailed mathematical model than used above. It is based on a pilot batch column given in [13] to separate a mixture of cyclohexane, n-haptane and toluene. The process consists of a large reboiler, a column with 18 trays, a total condenser and four distillate accumulators (Fig. 4). Two light fractions (cyclohexane and n-heptane) will be accumulated in the two main cut accumulators, respectively, during the batch. At the end of the batch the third product (toluene) will be received from the reboiler residue.

The model used to describe the process includes the dynamic component and energy balances and vapor-liquid equilibrium relations for each tray. Tray holdup, pressure and Murphree efficiency are taken into consideration. The differential-algebraic system formulated consists of 185 equations. Comparison of the simulation results in [13] to the experimental data given by NAD and SPIEGEL [18] demonstrate that the model is suitable for description of the real plant. For the numerical simulation, the program DOSOK (see [13, 14]) developed by P. LI was used.

The objective functional to be maximized (maximal profit) has the form

$$[c_1P_1 + c_2P_2 + c_3S(t_F) - c_4S(t_0)]/t_F - c_5$$

with certain positive constants $c_1, \dots, c_5$. As above, P_j describes the gain. The constants c_1, c_2

represents different prices of the products. S represents the contents of the reboiler. So $S(t_F)$ is essentially the third component remaining in the column while $S(t_0)$ is the feed amount. Purity conditions g_k are given as in the previous model. Additionally, a fourth interval $[t_3, t_F]$ is added such that an additional purity constraint on the third component, namely

$$g_3 = x_{3m}(t_F) \geq b_3,$$

is fulfilled.

The reflux policy was again approximated by a piecewise constant function. The whole process was split into four main phases $[t_{i-1}, t_i]$ with $t_4 = t_F$. These main sections were split further on into 3, 2, 2, and 1 subintervals, respectively, where $R(t)$ is held constant. The initial value solver uses a total of 18 subintervals. The dimension of the optimization problem is, therefore, 16.

Typical simulation results are given in Figures 5 and 6. Since the solver was implemented as a minimization problem, the negative of the objective function was minimized. In order to show the ability to find initial values which fulfill the purity constraints the initial data for the whole simulation were not chosen to be very well adjusted. The switching points as well as the reflux ratios are given in Table 2. The initial reflux policy (i.e. before optimization) and the final reflux policy are plotted in Fig. 7. The values of the objective functional are -3.27 before optimization and 3.55 afterwards. So the system starts with a loss in the initial configuration. The optimal value obtained is less than that of [14] (4.68) because we restricted the number of degrees of freedom to 16 compared to 80 in [14]. Figures 8 to 10 show the computed optimal trajectories of the concentrations and temperatures in the condenser and the reboiler.

This example was run on a Sun Ultra 1 Creator running at 169 MHz. The computation time for one simulation run is about 6 s. So the computation of a consistent strategy took about 18 min while the optimization took 40 min.

	break points in time (s)	reflux coefficient
Initial guess	0	10.0000
	3600	10.0000
	8100	10.0000
	10908	10.0000
	16488	10.0000
	19000	10.0000
	21500	10.0000
	24264	10.0000
	25740	
Consistent values	0	8.9122
	4044	9.4526
	9037	9.2233
	12845	8.2225
	18769	10.2090
	21850	8.7012
	24670	8.4219
	28155	9.4176
	30498	
Final values	0	7.8795
	3714	6.1483
	8231	8.4352
	11498	8.5879
	16189	10.6685
	18379	7.2928
	20852	7.0031
	24870	9.3652
	26866	

Table 2: Reflux policy

3.3 Model `m402`

This example was communicated to the authors by B. HEGNER as an input script for the simulation package `CHEMASIM` [6]. It describes the distillation process of a mixture of five components. The distillation column has 20 trays. Moreover, a reboiler, a condenser, and a number of other devices are attached to the column.

The model has some special features: The control uses the reflux ratio during the first and third fraction while the temperature is used in the second fraction. Moreover, the vapor/liquid equilibrium (K_{ip}) is strongly nonlinear.

Again, the reflux policy (including switching times) has to be chosen in order to maximize the product and to minimize the batch time. Plots of the iteration history are given in Figures 11 and 12. The optimization history shows that the annealing strategy should be reconsidered. Ideally, the acceptance ratio should approach 75%. This is roughly true until the 220th iterate. The next recomputation of the matrix Q leads to a large trial domain such that a lot of steps are rejected. Earlier changes of Q do not exhibit such an behavior. Even more, they are doing "the right thing". This behavior needs more investigation.

4 Conclusions

We applied a variant of simulated annealing to the optimization of batch distillation processes. The main emphasis was the use of *existing* production software. We aimed at the construction of an algorithm which can be easily implemented in production modeling environments. The simulation results demonstrate the performance and usability of the proposed approach. To our knowledge, reports on using simulated annealing have not been found.

Besides, two special properties are important to be mentioned: The problem of consistent initial data, i.e. data which give rise to states which fulfill mass and purity constraints, can easily be solved by the same algorithm. The overall efficiency of this approach has appeared to be significantly better than the penalty method. A reason might be that the masses and purities depend very sensitive on the control parameters. The set of admissible controls is usually very narrow in some directions of the parameter space.

The other nice property of simulated annealing is the fact that it converges towards a global minimum. However, the experience shows that while convergence is fast during the first steps it slows down considerably later on. So one should switch to a (locally but) over-linearly convergent method.

The problem of choosing optimal annealing parameters remains to be solved. At present, no general rule is available. The strange behavior of the annealing process shown in Fig. 12 emphasizes this fact.

Bibliography

[1] Ihor O. Bohachevsky, Mark E. Johnson, and Myron L. Stein. Generalized simulated annealing for function optimization. *Technometrics*, 28(3):209–217, 1986.

[2] W.B. Dolan, P.T. Cummings, and M.D. Levan. Process optimization via simulated annealing. *AIChE J.*, 35:725–736, 1989.

[3] John W. Eaton. *Gnu Octave – A high-level interactive language for numerical computations*. University of Wisconsin, Dept. of Chem. Eng., http://www.che.wisc.edu/octave, 2.0.12 edition, 1998.

[4] S. Farhat, M. Czernicki, L. Pibouleau, and S. Domenech. Optimization of multiple-fraction batch destillation by nonlinear programming. *AIChE J.*, 36(9):1349–1360, 1990.

[5] P. Floquet, L. Pibouleau, and S. Domenech. Separation sequence synthesis: How to use the simulated annealing procedure. *Comput. Chem. Eng.*, 18:1141–1148, 1994.

[6] B. Hegner. *CHEMASIM*. BASF AG Ludwigshafen.

[7] M.L. Huber. Structural optimization of vapor pressure correlations using simulated annealing and threshold accepting. *Comput. Chem. Eng.*, 18:929–932, 1994.

[8] John H. Kalivas, editor. *Adaption of simulated annealing to chemical optimization problems*. Number 15 in Data Handling in Science and Technology. Elsevier, Amsterdam, 1995.

[9] Armen Khatchaturyan. Statistical mechanics approach in minimizing a multivariate function. *J. Math. Phys.*, 27(7):1834–1838, 1986.

[10] S. Kirkpatrick, C. D. Gelatt jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[11] H.M. Ku and I.A. Karimi. An evaluation of simulated annealing for batch process scheduling. *Ind. Eng. Chem. Res.*, 30:163–169, 1991.

[12] Vladimír Kvasnička and Jiři Pospíchal. Simulated annealing. *Match*, 34:7–49, 10 1996.

[13] P. Li, H. P. Hoo, and G. Wozny. Efficient simulation of batch destillation processes by using orthogonal collocation. In *AIChE Annual Meeting*, 1997. Paper 204g.

[14] Pu Li and Günther Wozny. Dynamische Optimierung großer chemischer Prozesse mit Kollokationsverfahren am Beispiel Batch-Destillation. *Automatisierungstechnik*, 45(3):136–143, 1997.

[15] Ping Lu and M. Asif Khan. Nonsmooth trajectory optimization: An approach using continuous simulated annealing. *J. Guidance Contr. Dynamics*, 17(4):685–691, 1994.

[16] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087–1092, 1953.

[17] M.F.Cardoso, R.L.Salsedo, F. de Azevedo, and D.A. Barbosa. Simulated annealing approach to the aolution of minlp problems. *Comput. Chem. Eng.*, 21:1349–1364, 1997.

[18] M. Nad and L. Spiegel. Simulation of batch destillation by computer and comparision with experiment. In *The Use of Computers in Chemical Engeneering, Sicily, Italy*, page 737, 1987.

[19] J.D. Perkins and S.P.K. Walsh. Optimization as a tool in design/control integration. *Comp. Chem. Eng.*, 20:315–323, 1996.

[20] H. Edwin Romeijn and Robert L. Smith. Simulated annealing for constraint global optimization. *J. Global Optim.*, 5:101–126, 1994.

[21] B. E. Stuckman and P. L. Stuckman. Finding the "best" optimal control using global search. *Computers Elect. Engng.*, 19(1):9–18, 1993.

[22] P. J. M. van Laarhoven and E. H. L. Aarts. *Simulated Annealing: Theory and Applications*. Mathematics and Its Applications. D. Reidel, Dordrecht, 1988.

[23] David Vanderbilt and Steven G. Louie. A Monte Carlo simulated annealing approach to optimization over continuous variables. *J. Comput. Phys.*, 56:259–271, 1984.

[24] V. Černy. Thermodynamical approach to the traveling salesman problem. *J. Opt. Theory Appl.*, 45:41–51, 1985.

List of Figures

Appendix. Figures

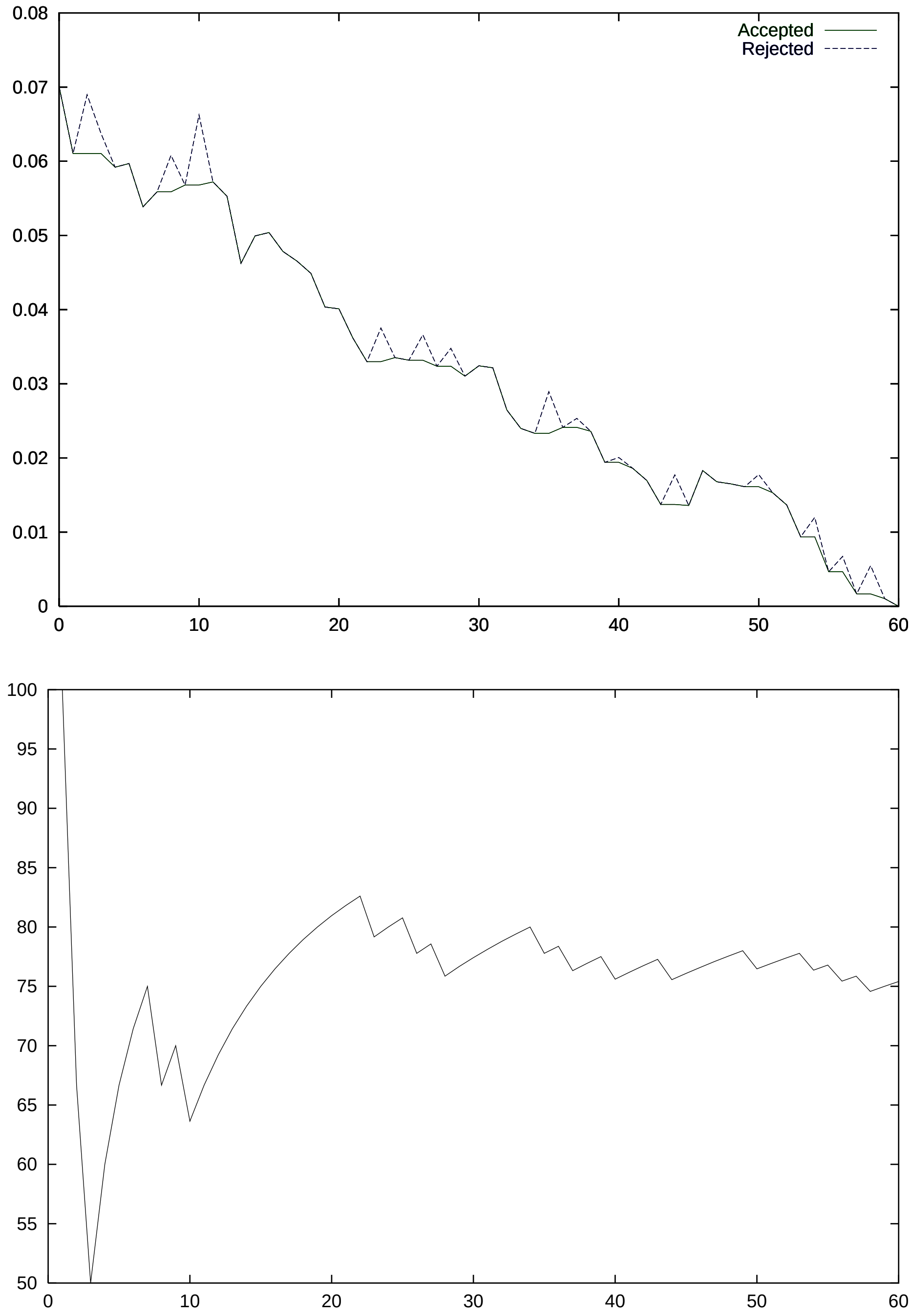

Figure 2: Computation of consistent initial values in S. FARHAT's model: annealing history (above) and acceptance ration (below).

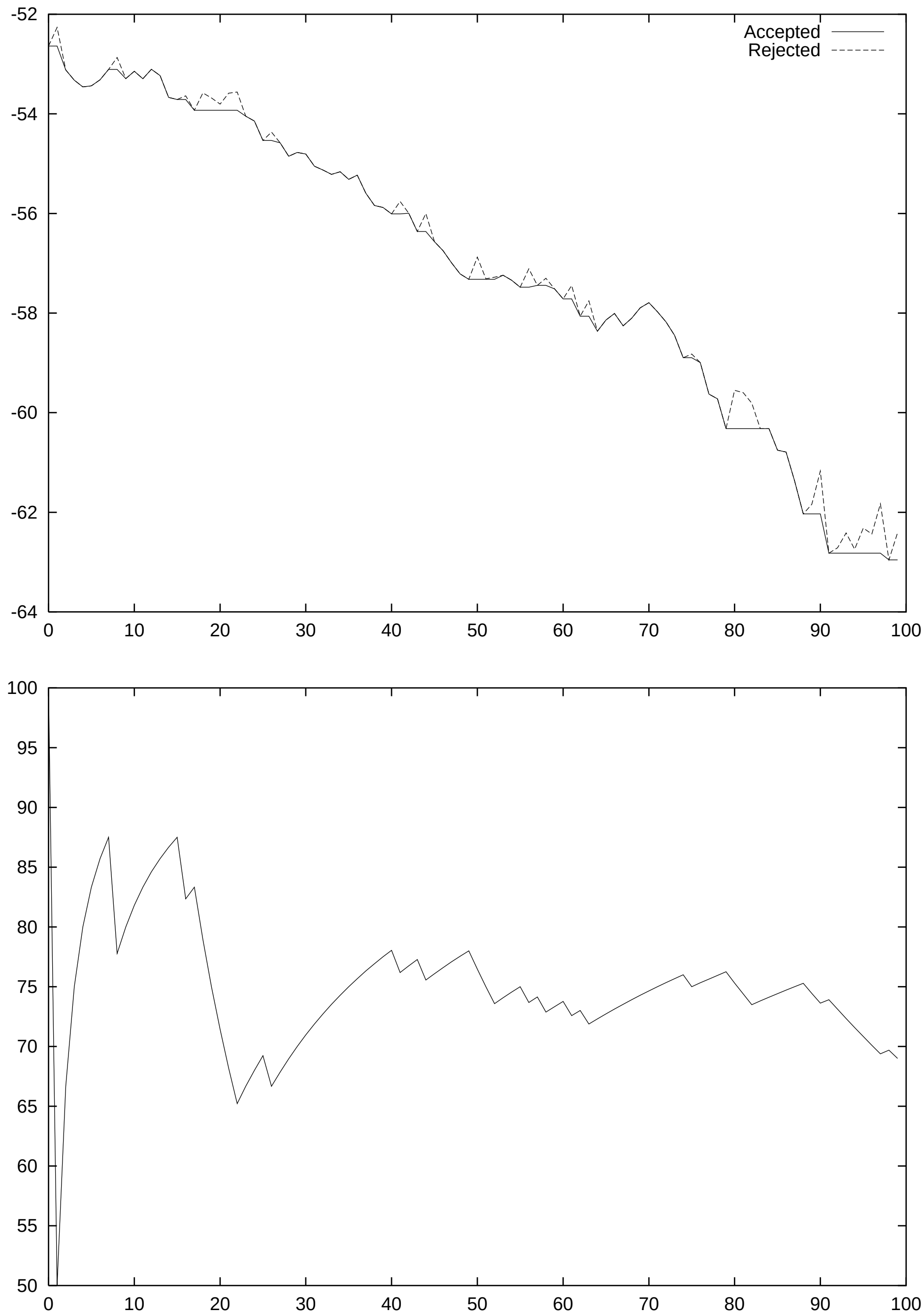

Figure 3: Optimization in S. FARHAT's model: annealing history (above) and acceptance ration (below).

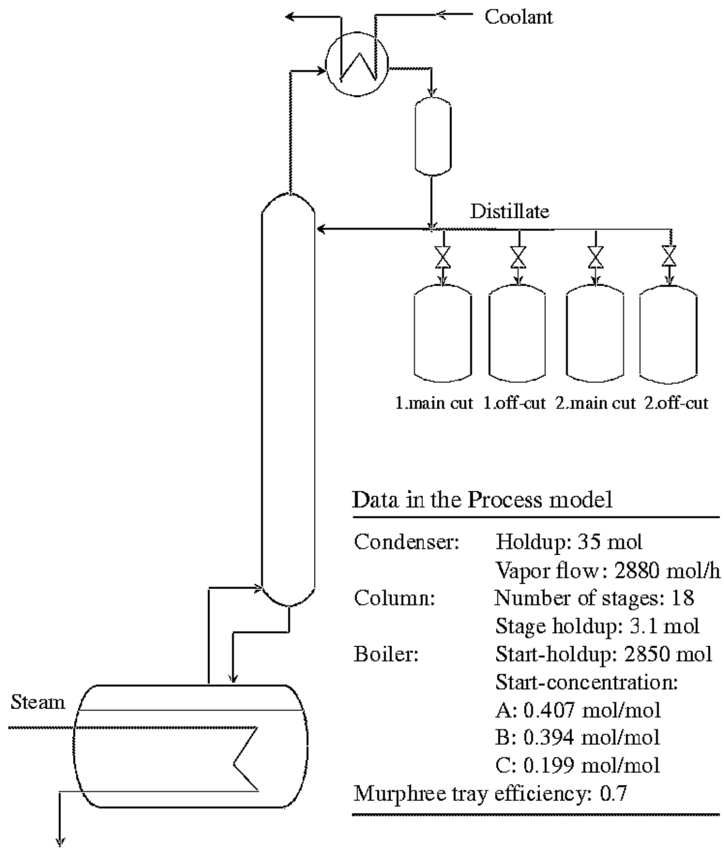

Figure 4: A pilot batch distillation process

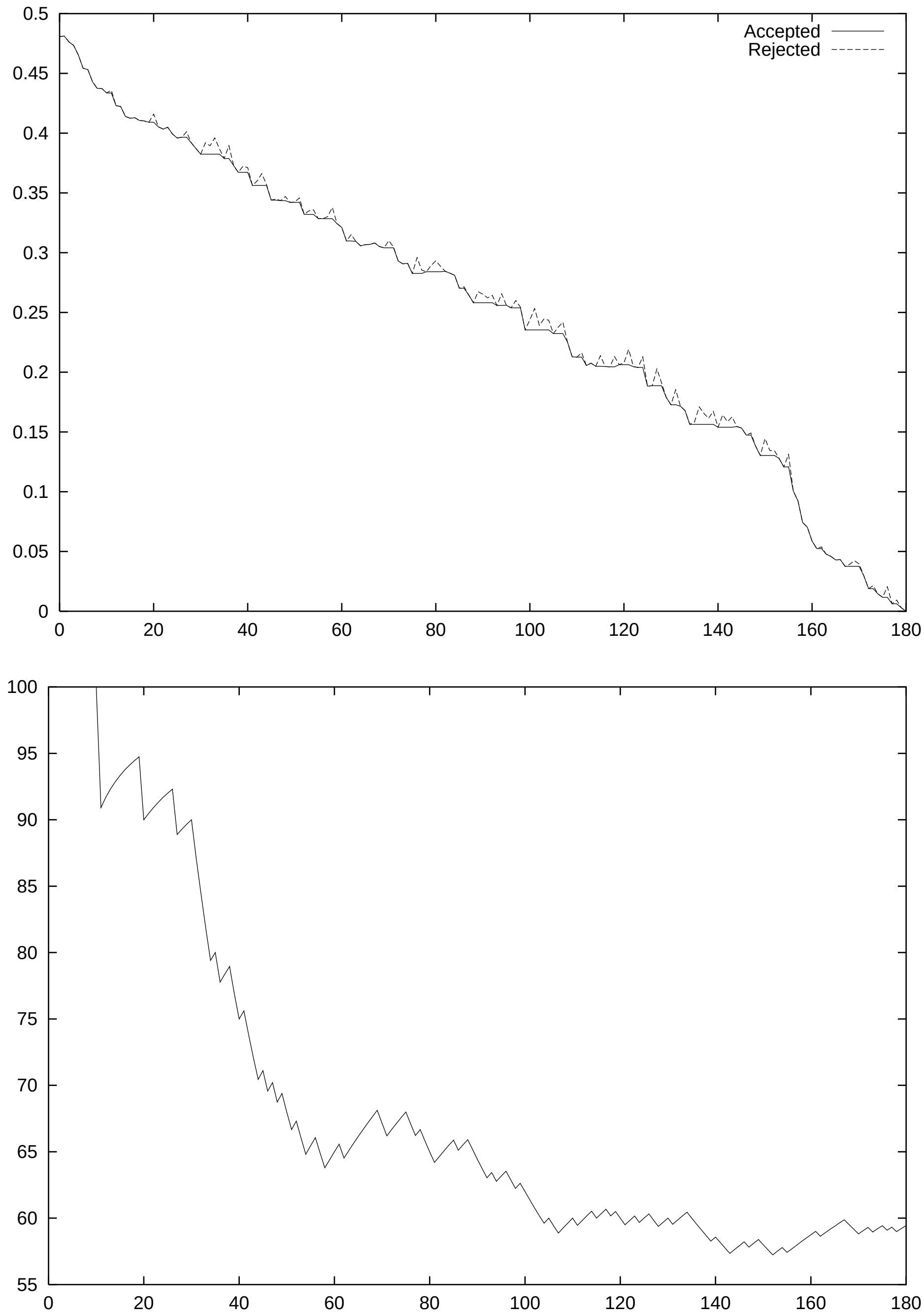

Figure 5: Computation of consistent initial values in the pilot batch distillation model: annealing history (above) and acceptance ration (below).

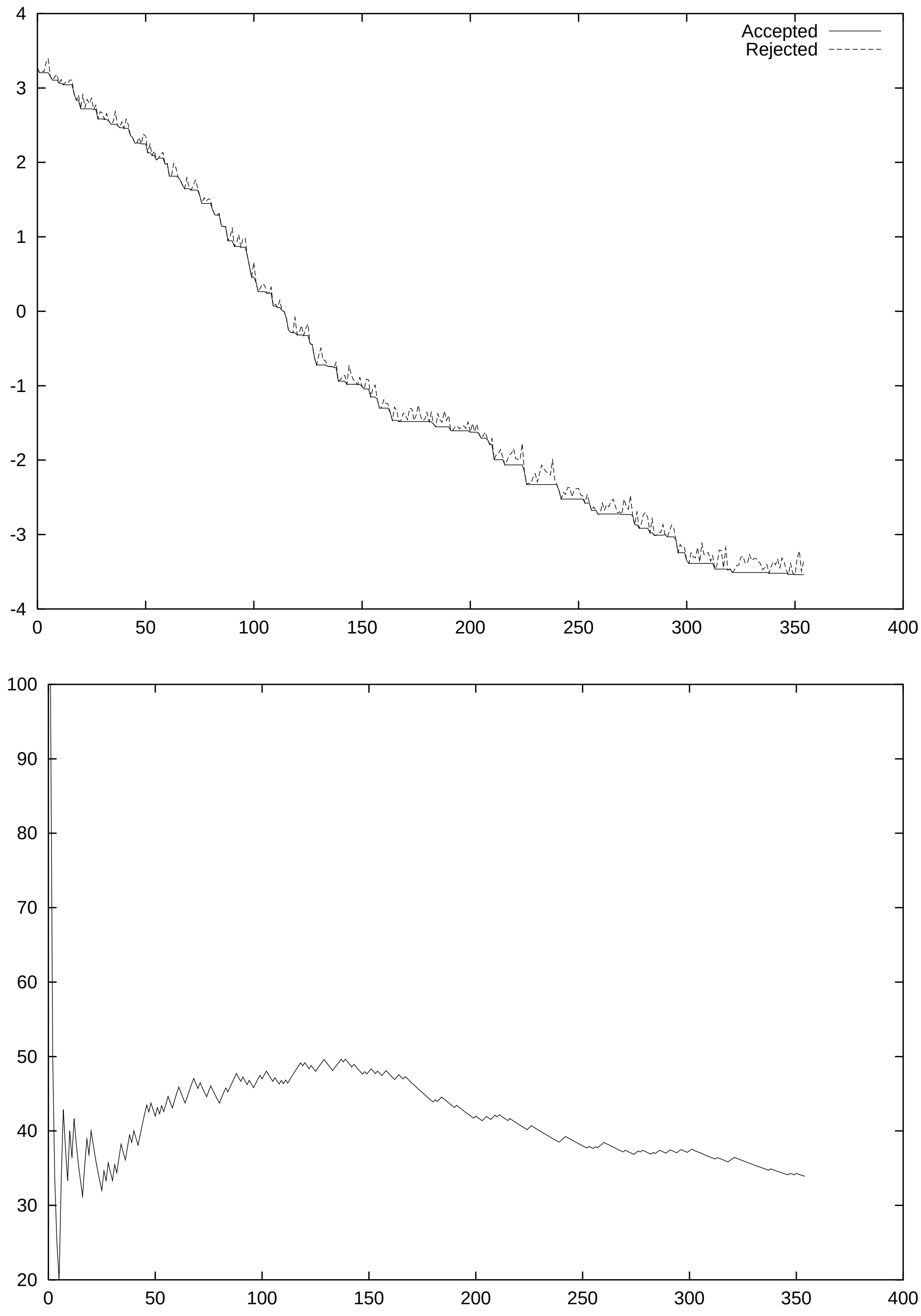

Figure 6: Optimization in the pilot batch distillation model: annealing history (above) and acceptance ration (below).

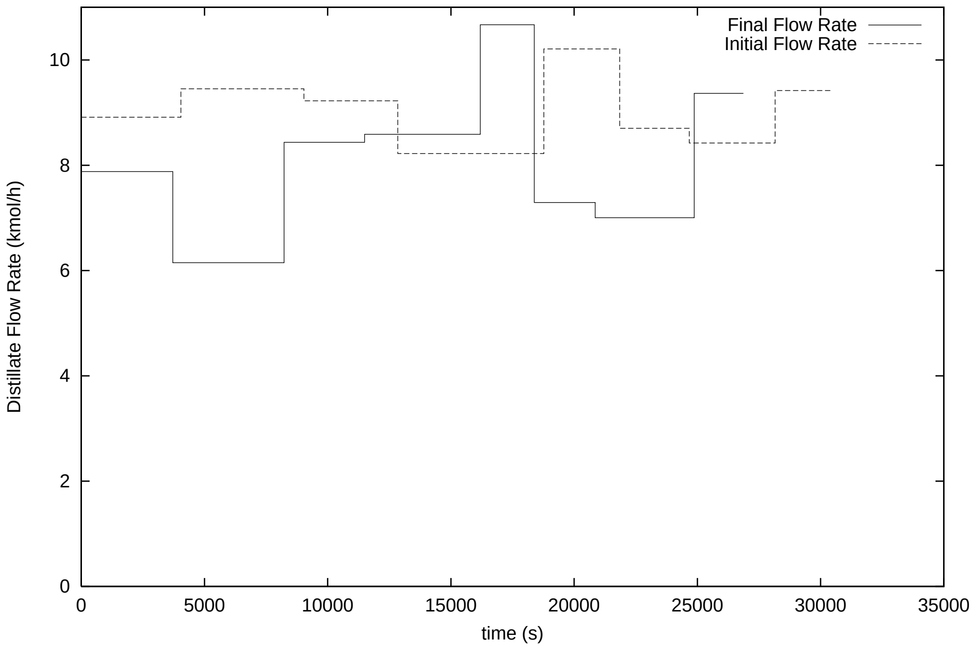

Figure 7: Pilot batch distillation model: Comparison of initial and final flux policies

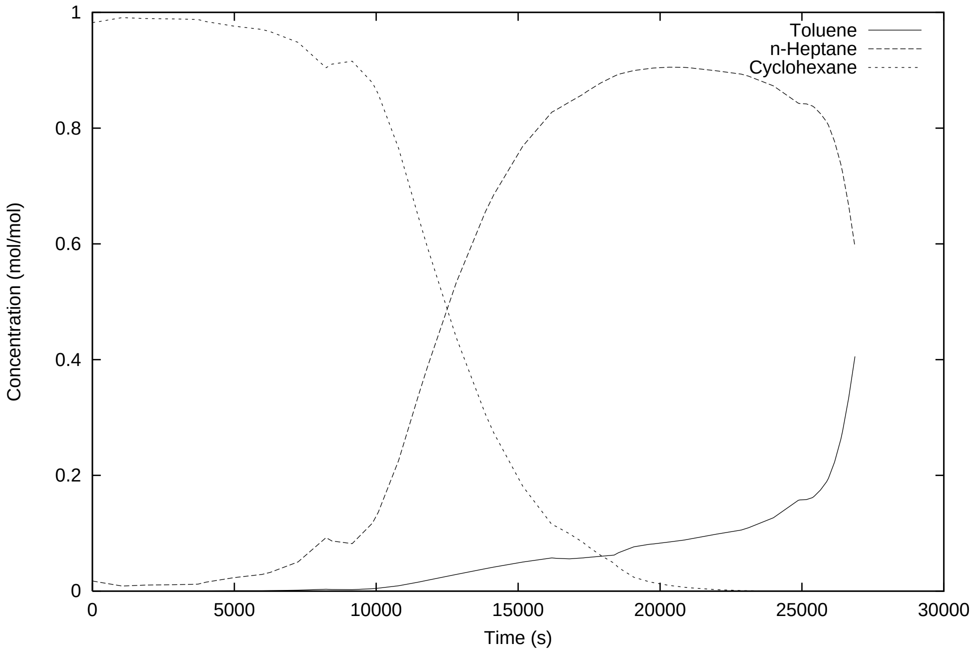

Figure 8: Pilot batch distillation model: Optimal concentration profiles of the condenser

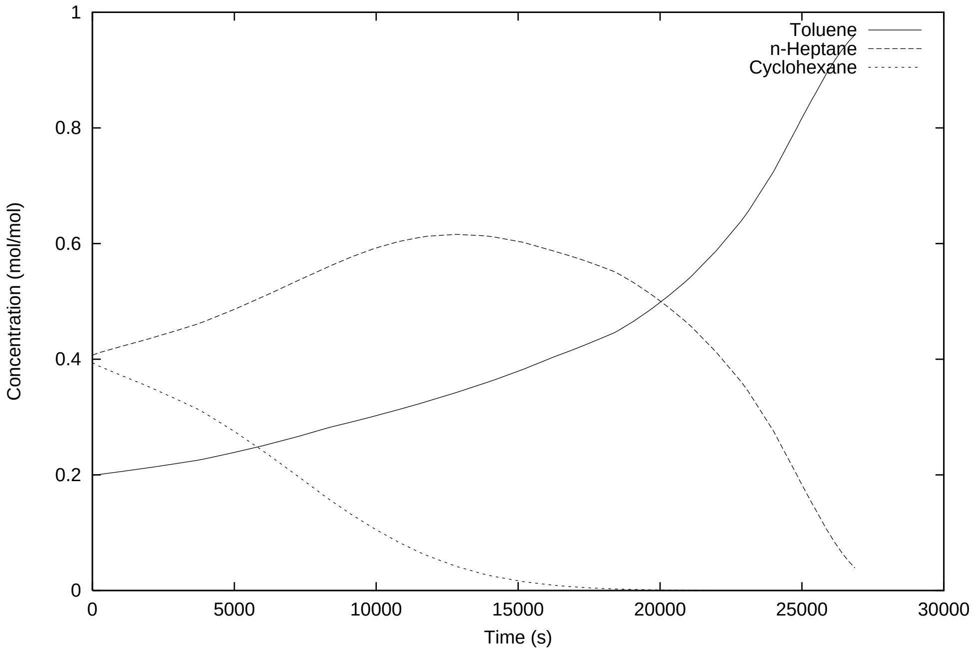

Figure 9: Pilot batch distillation model: Optimal concentration profiles of the reboiler

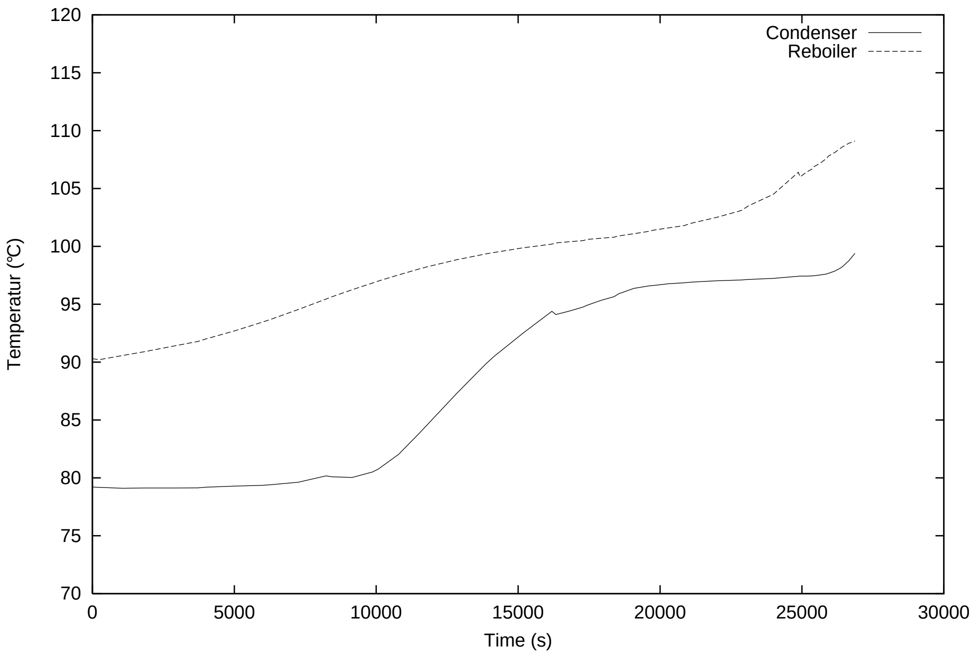

Figure 10: Pilot batch distillation model: Optimal temperature profiles of the condenser and the reboiler

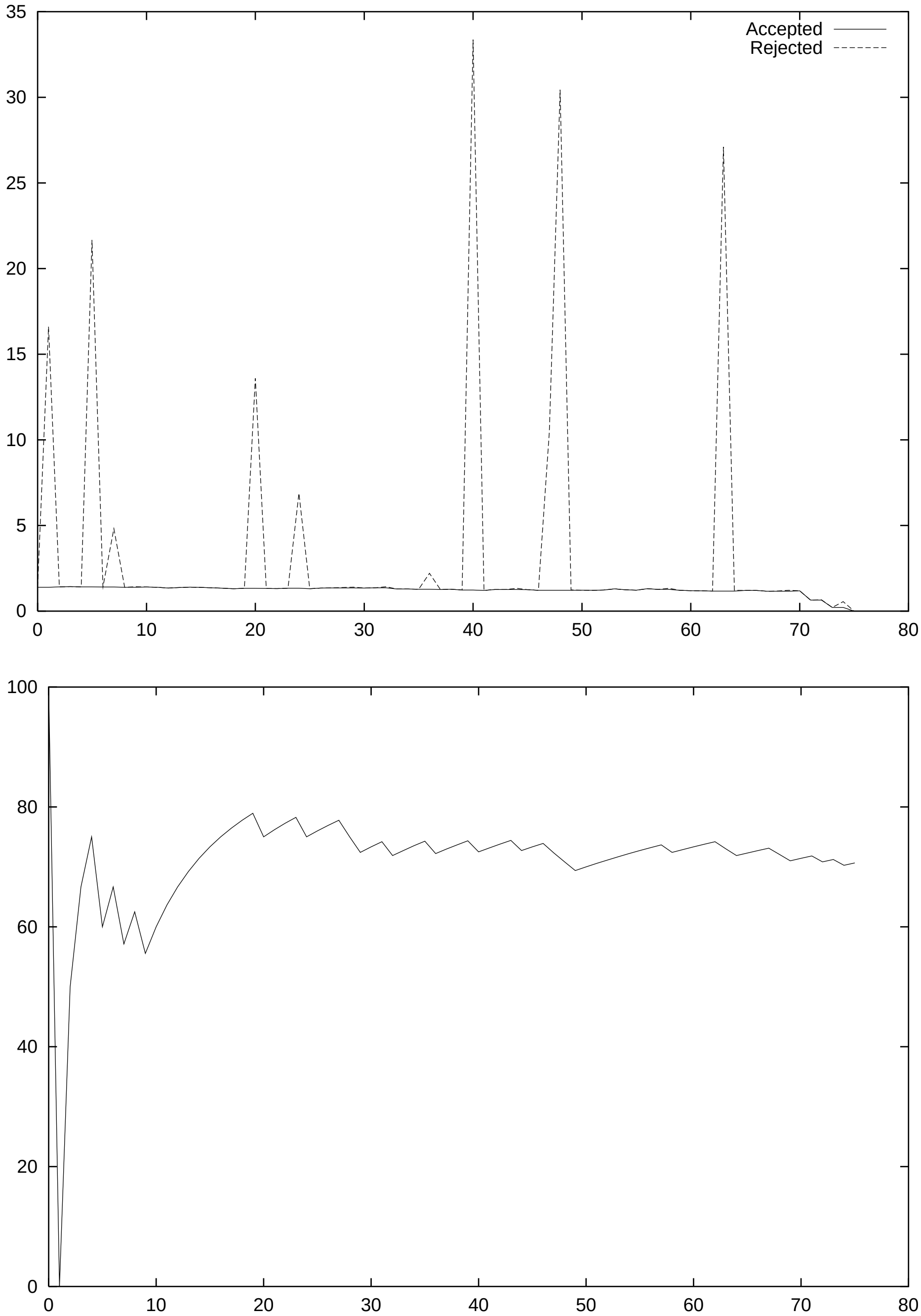

Figure 11: Computation of consistent initial values in `m402`: annealing history (above) and acceptance ration (below).

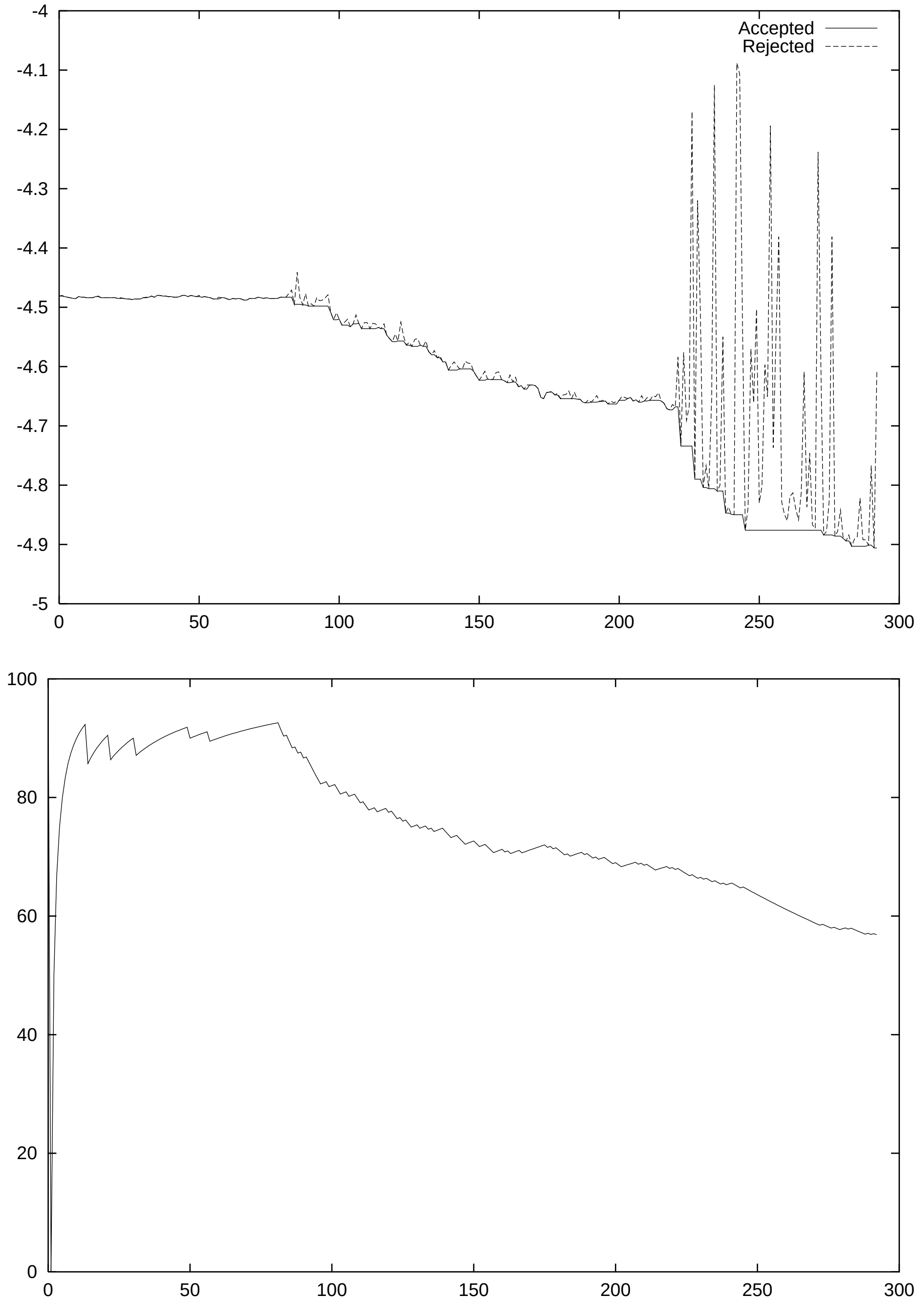

Figure 12: Optimization in `m402`: annealing history (above) and acceptance ration (below).