Optimization of the Batch Reactor by Means of Chaos Driven Differential Evolution

Roman Senkerik[1], Donald Davendra[2], Ivan Zelinka[2],
Zuzana Oplatkova[1], and Michal Pluhacek[1]

[1] Tomas Bata University in Zlin,
Faculty of Applied Informatics,
T.G. Masaryka 5555, 760 01 Zlin, Czech Republic
{senkerik,oplatkova,pluhacek}@fai.utb.cz
[2] VŠB-Technical University of Ostrava,
Faculty of Electrical Engineering and Computer Science,
Department of Computer Science, 17. listopadu 15,
708 33 Ostrava-Poruba, Czech Republic
{donald.davendra,ivan.zelinka}@vsb.cz

Abstract. In this paper, Differential Evolution (DE) is used in the task of optimization of a batch reactor. The novality of the approach is that a discrete chaotic dissipative standard map is used as the chaotic number generator to drive the mutation and crossover process in DE. The results obtained are compared with original reactor geometry and process parameters adjustment.

1 Introduction

These days the methods based on soft computing such as neural networks, evolutionary algorithms, fuzzy logic, and genetic programming are known as powerful tool for almost any difficult and complex optimization problem. They are often applied in engineering design problems [1] or for the purpose of optimization of processing plants or technological devices [2] - [5]. Chemical industry produces a whole range of products through chemical reactions. Generally, it may be stated that key technological points are chemical reactors. Designing optimal reactor parameters including control constitutes is one of the most complex tasks in process engineering.

This paper is aimed at investigating the chaos driven DE. Although a number of DE variants have been recently developed, the focus of this paper is the embedding of chaotic systems in the form of chaos number generator for DE and its application to optimization of batch reactor.

This research is an extension and continuation of the previous successful initial application based experiment with chaos driven DE [6].

The chaotic systems of interest are discrete dissipative systems. The Dissipative standard map was selected as the chaos number generator for DE.

Firstly, batch processes are explained. The next sections are focused on the description of the batch reactor, differential evolution, used chaotic systems and problem design. Results and conclusion follow afterwards.

V. Snasel et al. (Eds.): SOCO Models in Industrial & Environmental Appl., AISC 188, pp. 93–102.
springerlink.com

2 Characteristics of the Batch Processes

The optimization of batch processes has attracted attention in recent years [7], [8]. Batch and semi-batch processes are of considerable importance in the fine chemicals industry. A wide variety of special chemicals, pharmaceutical products, and certain types of polymers are manufactured in batch operations. Batch processes are typically used when the production volumes are low, when isolation is required for reasons of sterility or safety, and when the materials involved are difficult to handle. In batch operations, all the reactants are charged in a tank initially and processed according to a pre-determined course of action during which no material is added or removed. From a process systems point of view, the key feature that differentiates continuous processes from batch and semi-batch processes is that continuous processes have a steady state, whereas batch and semi-batch processes do not [9], [10].

Lot of modern control methods for chemical reactors were developed embracing the approaches such as iterative learning model predictive control [11], iterative learning dual-mode control [12] or adaptive exact linearization by means of sigma-point kalman filter [13]. Also the fuzzy approach is relatively often used [14]. Finally the methods of artificial intelligence are very frequently discussed and used. Many papers present successful utilization of either neural networks [15] - [17] or genetic (evolutionary) algorithms [18] - [20].

This paper presents the static optimization of the batch reactor by means of chaos driven differential evolution.

3 Description of the Reactor

This work uses a mathematical model of the reactor shown in Fig. 1. The reactor has two physical inputs (one for chemical substances and one for cooling medium) and one output (cooling medium).

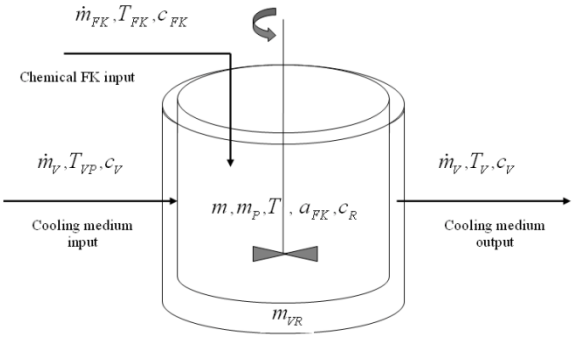

Fig. 1. Batch reactor

Chemical FK (**Filter Cake**) flows into the reactor through the input denoted "Chemical FK input", with parameters temperature-T_{FK}, mass flow rate-$\dot{m}_{FK}$ and specific heat-c_{FK}. The coolant flows into reactor through the second input denoted "Cooling medium", which is usually water of temperature T_{VP}, mass flow rate-$\dot{m}_V$ and specific heat-c_V.

Cooling medium flows through the jacket inner space of the reactor, with volume related to mass-m_{VR}, and flows out through the second output, with parameters mass flow rate m_V, temperature-T_V and specific heat-c_V.

At the beginning of the process there is an initial batch inside the reactor with parameter mass-m_P. The chemical FK is then added to this initial batch, so the reaction mixture inside the reactor has total mass-m, temperature-T and specific heat-c_R, and also contains partially unreacted portions of chemical FK described by parameter concentration a_{FK}.

This technique partially allows controlling the temperature of reaction mixture by the controlled feeding of the input chemical FK.

The main objective of optimization is to achieve the processing of large amount of chemical FK in a very short time. In general, this reaction is highly exothermal, thus the most important parameter is the temperature of the reaction mixture. **This temperature must not exceed 100°C** because of safety aspects and quality of the product. The original design of the reactor was based on standard chemical-technological methods and gives a proposal of reactor physical dimensions and parameters of chemical substances. These values are called within this paper **original parameters**.

Description of the reactor applies a system of four balance equations (1) and one equation (2) representing the term "k".

$$\dot{m}_{FK} = m'[t]$$

$$\dot{m}_{FK} = m[t]\, a'_{FK}[t] + k\, m[t] a_{FK}[t]$$

$$\dot{m}_{FK} c_{FK} T_{FK} + \Delta H_r k\, m[t] a_{FK}[t] = K\, S\,(T[t] - T_V[t]) + m[t] c_R\, T'[t]$$

$$\dot{m}_V c_V T_{VP} + K\, S\,(T[t] - T_V[t]) = \dot{m}_V c_V T_V[t] + m_{VR} c_V\, T'_V[t] \tag{1}$$

$$k = A e^{-\frac{E}{RT[t]}} \tag{2}$$

4 Differential Evolution

DE is a population-based optimization method that works on real-number-coded individuals [21]. A schematic is given in Fig. 2.

There are essentially five sections to the code. Section 1 describes the input to the heuristic. D is the size of the problem, G_{max} is the maximum number of generations, NP is the total number of solutions, F is the scaling factor of the solution and CR is

the factor for crossover. *F* and *CR* together make the internal tuning parameters for the heuristic.

Section 2 outlines the initialization of the heuristic. Each solution $x_{i,j,G=0}$ is created randomly between the two bounds $x^{(lo)}$ and $x^{(hi)}$. The parameter *j* represents the index to the values within the solution and *i* indexes the solutions within the population. So, to illustrate, $x_{4,2,0}$ represents the fourth value of the second solution at the initial generation.

After initialization, the population is subjected to repeated iterations in section 3. Section 4 describes the conversion routines of DE. Initially, three random numbers r_1, r_2, r_3 are selected, unique to each other and to the current indexed solution *i* in the population in 4.1. Henceforth, a new index j_{rand} is selected in the solution. j_{rand} points to the value being modified in the solution as given in 4.2. In 4.3, two solutions, $x_{j,r1,G}$ and $x_{j,r2,G}$ are selected through the index r_1 and r_2 and their values subtracted. This value is then multiplied by *F*, the predefined scaling factor. This is added to the value indexed by r_3.

However, this solution is not arbitrarily accepted in the solution. A new random number is generated, and if this random number is less than the value of *CR*, then the new value replaces the old value in the current solution. The fitness of the resulting solution, referred to as a perturbed vector $u_{j,i,G}$., is then compared with the fitness of $x_{j,i,G}$. If the fitness of $u_{j,i,G}$ is greater than the fitness of $x_{j,i,G}$., then $x_{j,i,G}$. is replaced with $u_{j,i,G}$; otherwise, $x_{j,i,G}$. remains in the population as $x_{j,i,G+1}$. Hence the competition is only between the new *child* solution and its *parent* solution.

Description of the used DERand1Bin strategy is presented in (3). Please refer to [21] and [22] for the detailed complete description of all other strategies.

$$u_{j,i,G+1} = x_{j,r1,G} + F \bullet \left(x_{j,r2,G} - x_{j,r3,G}\right) \tag{3}$$

1. Input: $D, G_{\max}, NP \geq 4, F \in (0,1+), CR \in [0,1]$, and initial bounds: $\vec{x}^{(lo)}, \vec{x}^{(hi)}$.

2. Initialize: $\begin{cases} \forall i \leq NP \wedge \forall j \leq D : x_{i,j,G=0} = x_j^{(lo)} + rand_j[0,1] \bullet \left(x_j^{(hi)} - x_j^{(lo)}\right) \\ i = \{1,2,...,NP\},\ j = \{1,2,...,D\},\ G = 0,\ rand_j[0,1] \in [0,1] \end{cases}$

$\begin{cases} \text{3. While } G < G_{\max} \\ \quad \forall i \leq NP \begin{cases} \text{4. Mutate and recombine:} \\ \qquad \text{4.1 } r_1, r_2, r_3 \in \{1,2,....,NP\}, \text{ randomly selected, except: } r_1 \neq r_2 \neq r_3 \neq i \\ \qquad \text{4.2 } j_{rand} \in \{1,2,...,D\}, \text{ randomly selected once each } i \\ \qquad \text{4.3 } \forall j \leq D, u_{j,i,G+1} = \begin{cases} x_{j,r_3,G} + F \cdot (x_{j,r_1,G} - x_{j,r_2,G}) \\ \qquad \text{if } (rand_j[0,1] < CR \vee j = j_{rand}) \\ x_{j,i,G} \quad \text{otherwise} \end{cases} \\ \text{5. Select} \\ \vec{x}_{i,G+1} = \begin{cases} \vec{u}_{i,G+1} & \text{if } f(\vec{u}_{i,G+1}) \leq f(\vec{x}_{i,G}) \\ \vec{x}_{i,G} & \text{otherwise} \end{cases} \end{cases} \\ G = G + 1 \end{cases}$

Fig. 2. DE Shematic

5 Chaotic Maps

This section contains the description discrete chaotic maps used as the random generator for DE. Iterations of the chaotic maps were used for the generation of real numbers in the process of crossover based on the user defined *CR* value and for the generation of the integer values used for selection of individuals [23].

5.1 Dissipative Standard Map

The Dissipative Standard map is a two-dimensional chaotic map. The parameters used in this work are $b = 0.1$ and $k = 8.8$ as suggested in [24]. For these values, the system exhibits typical chaotic behavior and with this parameter setting it is used in the most research papers and other literature sources. The Dissipative standard map is given in Fig. 3. The map equations are given in Eq. 3 and 4.

$$X_{n+1} = X_n + Y_{n+1} (\mathrm{mod} 2\pi) \tag{3}$$

$$Y_{n+1} = bY_n + k \sin X_n (\mathrm{mod} 2\pi) \tag{4}$$

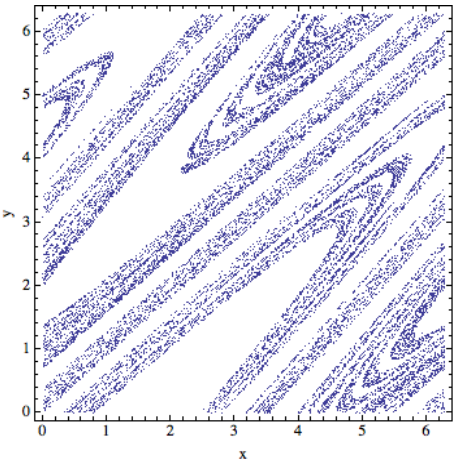

Fig. 3. Dissipative standard map

6 Problem Design

The cost function (CF) that was minimized is given in (5). It is divided into three time intervals, and has two penalizations. The first part ensures minimizing the area between required and actual temperature of the reaction mixture, and the second part ensures the rapid reaction of the whole batch of chemical FK, thus the very low value of concentration a_{FK} of partly unreacted portions of chemical FK in reaction mixture.

The first penalization helps to find solutions in which the temperature of reaction mixture cools down fast to its initial state, and the process duration is shortened. The second corresponds to the critical temperature.

$$f_{\cos t} = \sum_{t=0}^{t_1} |w - T[t]| + \sum_{t=0}^{t_1} a_{FK}[t] + pen.1 + pen.2 \tag{5}$$

$$pen.1 = \begin{cases} 0 & Max(T[\tau]) \le 323{,}15 \\ 50000 & else \end{cases}$$

for $\tau \in \langle t_2, t_3 \rangle$

$$pen.2 = \begin{cases} 0 & Max(T[\tau]) \le 373{,}15 \\ 50000 & else \end{cases}$$

for $\tau \in \langle 0, t_3 \rangle$

Where the time intervals were set for example as: $t_1 =$ 15000 s; $t_2 =$20000 s; $t_3 =$25000 s.

The minimizing term, presented in (6), limits the maximum mass of one batch. Moreover, many parameters were interrelated due to the optimization of the reactor geometry.

$$m[t] \le m_{\max} \tag{6}$$

7 Results

The parameter settings for ChaosDE were obtained analytically based on numerous experiments and simulations (see Table 2). Experiments were performed in an environment of Wolfram Mathematica and were repeated 50 times.

Table 1. Parameter set up for Chaos DE

Parameter	Value
PopSize	50
F	0.8
CR	0.8
Generations	200
Max. CF Evaluations (CFE)	10000

The optimization proceeded with the parameters shown in Table 2, where the internal radius of reactor is expressed in parameter *r* and is related to cooling area *S*. Parameter *d* represents the distance between the outer and inner jackets and parameter *h* means the height of the reactor.

Table 2. Optimized reactor parameters, difference between original and the optimized reactor

Parameter	Range	Original value	Optimized value
$\dot{m}_{FK}$ [kg.s^{-1}]	0 – 10.0	0 - 3	0,0696
T_{VP} [K]	273.15 – 323.15	293.15	276.077
$\dot{m}_V$ [kg]	0 – 10.0	1	4,6355
r [m]	0.5 – 2.5	0.78	1.0
h [m]	0.5 – 2.5	1.11	1.0
d [m]	0.01 – 0.2	0.03	0.0749

The design of the reactor was based on standard chemical-technological methods and gives a proposal of reactor physical dimensions and parameters of chemical substances. These values are called within this paper **original parameters (values)**. The best results of the optimization are shown in Fig.4 and Table 2. From these results it is obvious that the temperature of reaction mixture did not exceed the critical value. The maximum temperature was 372.8 K (99.65°C). The required temperature w used in cost function was 370.0 K (96.85°C). Another fact not to be neglected is the shortened duration of the process (approx 20740 s compared to the original approx 25000 s). Fig 4 shows the time evolution of CF value for the best individual solution, whereas Fig 5 confirms the robustness of ChaosDE in finding the best solutions for all 50 runs.

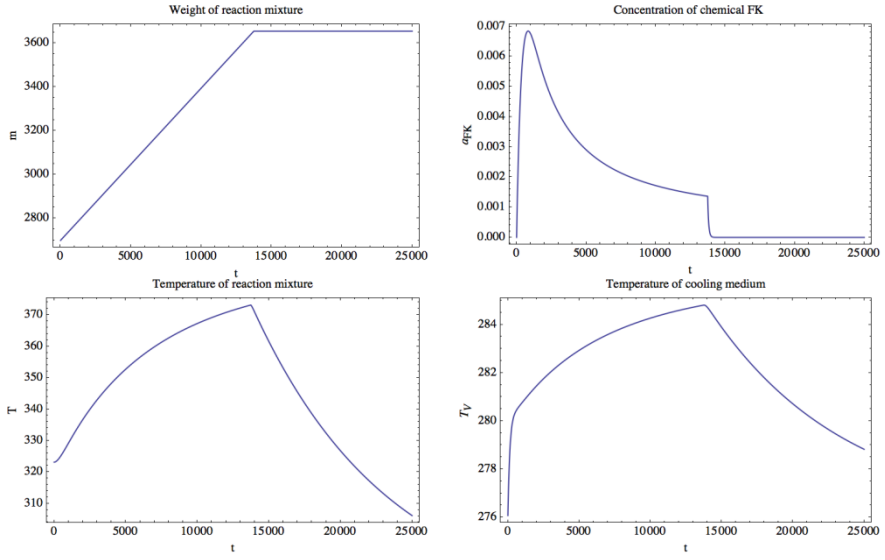

Fig. 4. Results of optimization, Simulation of the **best solution**, course of the weight of reaction mixture (upper left), concentration of chemical FK (upper right), temp. of reaction mixture (lower left), and temp. of cooling medium (lower right)

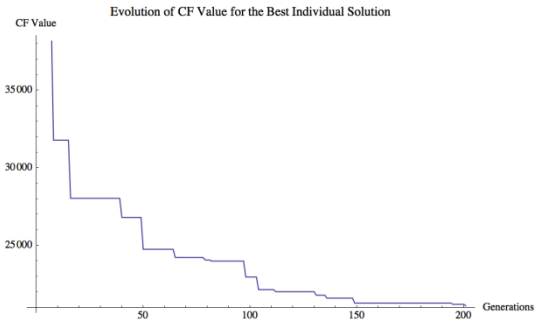

Fig. 5. Evolution of CF value for the best individual solution

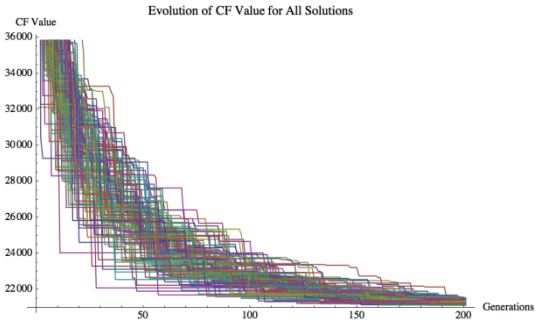

Fig. 6. Evolution of CF value for all 50 runs of Chaos-DE

8 Conclusions

Based on obtained results, it may be claimed, that the presented ChaosDE driven by means of the chaotic Dissipative standard map has given satisfactory results. The behavior of an uncontrolled original reactor gives quite unsatisfactory results. The reactor parameters found by optimization brought performance superior compared to the reactor set up by an original parameters.

Future plans include testing of different chaotic systems, comparison with different heuristics and obtaining a large number of results to perform statistical tests.

Acknowledgments. This work was supported by European Regional Development Fund under the project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089 and project IT4Innovations Centre of Excellence No. CZ.1.05/1.1.00/02.0070, and by Internal Grant Agency of Tomas Bata University under the project No. IGA/FAI/2012/037.

References

[1] He, Q., Wang, L.: An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Engineering Applications of Artificial Intelligence 20, 89–99 (2007)

[2] Silva, V.V.R., Khatib, W., Fleming, P.J.: Performance optimization of gas turbine engine. Engineering Applications of Artificial Intelligence 18, 575–583 (2005)

[3] Tan, W.W., Lu, F., Loh, A.P., Tan, K.C.: Modeling and control of a pilot pH plant using genetic algorithm. Engineering Applications of Artificial Intelligence 18, 485–494 (2005)

[4] Lepore, R., Wouwer, A.V., Remy, M., Findeisen, R., Nagy, Z., Allgöwer, F.: Optimization strategies for a MMA polymerization reactor. Computers and Chemical Engineering 31, 281–291 (2007)

[5] Rout, B.K., Mittal, R.K.: Optimal manipulator parameter tolerance selection using evolutionary optimization technique. Engineering Applications of Artificial Intelligence 21, 509–524 (2008)

[6] Davendra, D., Zelinka, I., Senkerik, R.: Chaos driven evolutionary algorithms for the task of PID control. Computers & Mathematics with Applications 60(4), 898–1221 (2010) ISSN 0898-1221

[7] Silva, C.M., Biscaia, E.C.: Genetic algorithm development for multi-objective optimization of batch free-radical polymerization reactors. Computers and Chemical Engineering 27, 1329–1344 (2003)

[8] Arpornwichanop, A., Kittisupakorn, P., Mujtaba, M.I.: On-line dynamic optimization and control strategy for improving the performance of batch reactors. Chemical Engineering and Processing 44(1), 101–114 (2005)

[9] Srinisavan, B., Palanki, S., Bonvin, D.: Dynamic optimization of batch processes I. Characterization of the nominal solution. Computers and Chemical Engineering 27, 1–26 (2002)

[10] Srinisavan, B., Palanki, S., Bonvin, D.: Dynamic optimization of batch processes II. Role of Measurement in Handling Uncertainly. Computers and Chemical Engineering 27, 27–44 (2002)

[11] Wang, Y., Zhou, D., Gao, F.: Iterative learning model predictive control for multi-phase batch processes. Journal of Process Control 18, 543–557 (2008)

[12] Cho, W., Edgar, T.F., Lee, J.: Iterative learning dual-mode control of exothermic batch reactors. Control Engineering Practice 16, 1244–1249 (2008)

[13] Beyer, M.A., Grote, W., Reinig, G.: Adaptive exact linearization control of batch polymerization reactors using a Sigma-Point Kalman Filter. Journal of Process Control 18, 663–675 (2008)

[14] Sarma, P.: Multivariable gain-scheduled fuzzy logic control of an exothermic reactor. Engineering Applications of Artificial Intelligence 14, 457–471 (2001)

[15] Sjöberg, J., Mukul, A.: Trajectory tracking in batch processes using neural controllers. Engineering Applications of Artificial Intelligence 15, 41–51 (2002)

[16] Mukherjee, A., Zhang, J.: A reliable multi-objective control strategy for batch processes based on bootstrap aggregated neural network models. Journal of Process Control 18, 720–734 (2008)

[17] Mujtaba, M., Aziz, N., Hussain, M.A.: Neural network based modelling and control in batch reactor. Chemical Engineering Research and Design 84(8), 635–644 (2006)

[18] Causa, J., Karer, G., Nunez, A., Saez, D., Skrjanc, I., Zupancic, B.: Hybrid fuzzy predictive control based on genetic algorithms for the temperature control of a batch reactor. Computers and Chemical Engineering (2008), doi:10.1016/j.compchemeng.2008.05.014

[19] Altinten, A., Ketevanlioglu, F., Erdogan, S., Hapoglu, H., Alpbaz, M.: Self-tuning PID control of jacketed batch polystyrene reactor using genetic algorithm. Chemical Engineering Journal 138, 490–497 (2008)
[20] Faber, R., Jockenhövel, T., Tsatsaronis, G.: Dynamic optimization with simulated annealing. Computers and Chemical Engineering 29, 273–290 (2005)
[21] Price, K.: An Introduction to Differential Evolution. In: Corne, D., Dorigo, M., Glover, F. (eds.) New Ideas in Optimization, pp. 79–108. McGraw-Hill, London (1999) ISBN 007-709506-5
[22] Price, K., Storn, R.: Differential evolution homepage (2001), http://www.icsi.berkeley.edu/~storn/code.html
[23] Caponetto, R., Fortuna, L., Fazzino, S., Xibilia, M.: Chaotic sequences to improve the performance of evolutionary algorithms. IEEE Trans. Evol. Comput. 7(3), 289–304 (2003)
[24] Sprott, J.C.: Chaos and Time-Series Analysis. Oxford University Press (2003)