CHEMICAL ENGINEERING TRANSACTIONS Volume 17, 2009
Editor Sauro Pierucci
 ISBN 978-88-95608-01-3 ISSN 1974-9791
DOI: 10.3303/CET0917228

Comparison of the Optimisation Performance of Particle Swarm Optimisation and Genetic Algorithms applied to a Three-Phase Slurry Catalytic Reactor

Mylene C. A. F. Rezende[1], Caliane B. B. Costa[1], Delba N. C. Melo[1], Adriano P. Mariano[1], Eduardo C. Vasco de Toledo[2], Rubens Maciel Filho[1]
[1]School of Chemical Engineering - University of Campinas, UNICAMP
P.O. Box 6066, Zip Code 13083-970, Campinas-SP, Brazil
E-mail address: mylene@lopca.feq.unicamp.br
[2]Petrobras SA, Paulínia Refinery (REPLAN)
Rodovia SP 332 - Km 132, P.O. Box 1, 13140-000. Paulínia, SP- Brazil.

Multiphase reactors are largely used in industrial processes like hydrogenation and oxidation with usually high throughput production. These types of industrial chemical processes are characterized by a complex dynamic behaviour that requires high dimensionality and non-linear mathematical models, being difficult to be optimised by conventional methods. In face of this, the present work aims to employ heuristic methods to optimise a type of multiphase reactor. The idea is to employ a Particle Swarm Optimisation (PSO) and to compare the performance of this method with the performance of Genetic Algorithms (GAs), when applied to a three-phase catalytic slurry reactor in which the reaction of the hydrogenation of o-cresol producing 2-methyl-cyclohexanol occurs. Heuristic optimisation methods have the advantages of not requiring manipulation of the mathematical structure of the objective function and/or constraint and not requiring an initial feasible point. PSO and GAs have been successfully applied to a range of problems and show characteristics of easiness of implementation and capability of escaping local optimal solution. In this way, in order to optimise the process, the PSO code has been coupled with the rigorous mathematical model of the reactor and the same has been done with GA code. The aim of the optimisation is the searching of the process inputs that maximize the productivity of 2-methyl-cyclohexanol subject to the environmental constraint of conversion. The comparison of the optimisation performance of both methods, PSO and GAs, can be done in relation to computational time, to objective function value and to facility of implementation, determining the suitable method for real-time optimisation.

1. Introduction

Hydrogenation multiphase reactors are typical to be large scale system with usually high throughput production so that even small increase in productivity or cost reduction may have a significant impact on process performance. Such type of reactors is characterized by a complex behaviour due to heat and mass interaction so that the mathematical

Please cite this article as: Rezende M.C., Costa C., Melo D., Mariano A., Vasco De Toledo E. and Maciel Filho R., (2009), Comparison of the optimization performance of particle swarm optimization and genetic algorithms applied to a three-phase slurry catalytic reactor, Chemical Engineering Transactions, 17, 1365-1370 DOI: 10.3303/CET0917228

modeling leads to a system of partial differential equations that needs to be solved numerically. The process optimisation requires a suitable solution procedure that allows optimal values to be found with accuracy and robustness, especially regarding the initial values. For that type of problem, solution procedure based on deterministic methods tends to fail. A reasonable approach is to make use of heuristic methods and among them Genetic Algorithms (GAs) and Particle Swarm Optimisation (PSO) are good candidates. Heuristic optimisation methods have the advantages of not requiring manipulation of the mathematical structure of the objective function and/or constraint and not requiring an initial feasible point. PSO and GAs have been successfully applied to a range of problems and have characteristics of easiness of implementation and capability of escaping local optimal solution (Zhang et al., 2006; Deb, 2000).
This paper proposes to employ a PSO and to compare the performance of this method with the performance of the GAs, when applied to a three-phase catalytic slurry reactor in which the reaction of the hydrogenation of o-cresol producing 2-methyl-cyclohexanol occurs. This comparison is done in relation to computational time, to objective function value and to facility of implementation, determining the suitable method for real-time optimisation.

2. Genetic Algorithms and Particle Swarm Optimisation

GAs are based on Natural Genetic and Natural Selection mechanism and some fundamental ideas are borrowed from Genetics in order to artificially construct an optimisation procedure. The GAs start with a population of possible solutions, which suffers evolution during the generations. Each solution is coded as a set of binary or real strings (chromosome), each string representing a variable in the solution. The evolution occurs when some genetic operators as reproduction, crossover and mutation are applied. The survival of the fittest is achieved by the assignment of a fitness function.
PSO imitates the social behaviour of organisms such as birds flocking and fish schooling. Each solution candidate of the optimisation problem (called particle) flies in the problem search space looking for the optimal position according to its own experience as well as to the experience of its neighbourhood. The performance of each particle is evaluated using a predefined fitness function, which capturers the characteristics of the optimisation problem (Zhang et al., 2006). The velocity and position of each particle are updated according to the following equations:

$$v_{j,g}^{(k+1)} = w.v_{j,g}^{(k)} + c_1.r_1.(pbest_{j,g} - x_{j,g}^{(k)}) + c_2.r_2.(gbest_g - x_{j,g}^{(k)}) \quad (1)$$

$$x_{j,g}^{(k+1)} = x_{j,g}^{(k)} + v_{j,g}^{(k+1)} \text{ , with } j = 1, 2, ..., n \text{ and } g = 1, 2, ... , m \quad (2)$$

where n = number of particles in the swarm; m = number of components (operating variables to be optimised) for the vectors v_j and x_j; k = number of iterations; $v_{j,g}^{(k)}$ = the g^{th} component of the velocity of particle j at iteration k; w = inertia weight factor; c_1 and c_2 = cognitive and social acceleration factors, respectively; r_1 and r_2 = random

numbers uniformly distributed in the range (0, 1); $x_{j,g}^{(k)}$ = the g^{th} component of the position of particle j at iteration k; $pbest_j$ = the best fitness of particle j; $gbest$ = the overall best out of all the particles in the population.
Each particle moves in the search space with a velocity according to its own previous best solution and its group's previous best solution. The velocity update in a PSO consists of three parts, momentum, cognitive and social parts, respectively, each term of the right side of Equation 1. The balance among these parts determines the performance of a PSO algorithm. The parameters c_1 and c_2 determine the relative pull of *pbest* and *gbest* and the parameters r_1 and r_2 help in stochastically varying these pulls (Panda and Padhy, 2008).

3. Optimisation Problem Formulation for the Hydrogenation Multiphase Reactor

In the present paper, the o-cresol hydrogenation carried out in a three-phase catalyst slurry reactor in order to obtain 2-methyl-cyclohexanol is used as study case. It is typical of many industrially important large scale chemical processes where even small improvement has significance in the economical performance. The process can be described by the reaction that follows, since side reactions have not been detected:

$$3\,H_{2(g)} + C_6H_4OHCH_{3(l)} \rightarrow C_6H_{10}OHCH_{3(l)} \tag{3}$$

The mathematical model of this reactor was developed by Vasco de Toledo et al. (2001) and the respective equations can be found in Rezende et al. (2008). The mathematical model is characterized by a high dimensionality and non-linearity. The optimising variables in this model are the linear velocity of gas (ug), the linear velocity of liquid (ul), the linear velocity of coolant (ur), the hydrogen concentration in the gas phase in the reactor feed (Agf), the hydrogen concentration in the liquid phase in the reactor feed (Alf), the o-cresol concentration in the liquid phase in the reactor feed (Blf), the feed reactor temperature (Tf) and the feed coolant temperature (Trf). The output variables are exit concentrations of hydrogen both in the gas phase (Ag) and in the liquid phase (Al), exit concentration of o-cresol in the liquid phase (Bl), reaction medium temperature at the exit of the reactor (T) and temperature of the coolant fluid at the exit of the reactor (Tr). The objective function is given by the maximization of the productivity of 2-methyl-cyclohexanol, as calculated by Equation 4:

$$f(x) = Productivity = \frac{(Blf - Bl) * ul}{L} \tag{4}$$

where L is the reactor length that is equal to two meters.
Since, the productivity of the process is deeply dependent on the o-cresol conversion, the o-cresol conversion is defined as a constraint of the process by Equation 5:

$$g_1(x) = Conversion = \frac{Blf - Bl}{Blf} > 0.90 \tag{5}$$

The optimisation problem can be written as in Equation 6:

$$\begin{aligned} &\max_{x} f = \max \text{ productivity} \\ &\text{subject to : Model equations} \\ &\qquad \text{Conversion} > 90\% \end{aligned} \tag{6}$$

where x is vector composed by the input variables (ug, ul, ur, Agf, Alf, Blf, Tf and Trf).

The lower and upper bounds of the optimisation variables of the vector x are set as shown in Table 1. These values are compatible with industrial size reactors and came from a model validation procedure (Vasco de Toledo et al., 2001).

Table 1 - Lower and upper bounds of the optimisation variables.

Lower bounds	Optimisation Variables	Upper bounds
4.195×10^{-3}	ul (m/s)	1.1805×10^{-2}
1.08	ug (m/s)	2.52
3.0×10^{-3}	ur (m/s)	7.0×10^{-3}
2.392×10^{-3}	Agf (kmol/m^3)	6.08×10^{-3}
7.5×10^{-4}	Alf (kmol/m^3)	2.25×10^{-3}
9.732×10^{-3}	Blf (kmol/m^3)	3.827×10^{-2}
459.0	Tf (K)	621.0
425.0	Trf (K)	575.0

In order to solve the constrained problem, a constraint handling method based on the penalty function approach was used not requiring any penalty parameter, since infeasible solutions are compared based only on their constraint violation (Deb, 2000). This method has been employed for both GA and PSO optimisation procedures. The expression of the fitness function (F(x)) for the maximisation problem, presented on Equation 6, is given by Equation 7:

$$F(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & \text{if } g_j(\mathbf{x}) \geq 0 \quad \forall j=1,2,...,nc \\ f_{\min} - \sum_{j=1}^{m} \left\langle g_j(\mathbf{x}) \right\rangle & \text{otherwise} \end{cases} \tag{7}$$

where $f_{\min}$ is the objective function value of the worst feasible solution in the population.

In order to optimise the reactor, a GA code and a PSO code are coupled with the non-linear mathematical model (Rezende et al., 2008). The used codes and the results obtained are shown in the next sections.

4. Optimisation of the reactor using GA and PSO

The GA code used in this work is the Fortran GA driver based on binary code, developed by Carroll (2008). In this work, the uniform crossover, jump and creep mutations, niching and elitism were used. The GA code requires a set-up of input

parameters that in this paper are the optimised parameters by factorial design approach developed by Rezende et al. (2008) indicated in Table 2. The PSO code used in this work is a Fortran PSO driver. Equations 1 and 2 contain some PSO input parameters that are also set on Table 2.

Table 2 - GA and PSO input parameters.

GA input parameters	Value
Population size per generation	50
Maximum number of generations	50
Crossover probability	0.70
Jump mutation probability	0.0248
Creep mutation probability	0.04
Initial random number seed for the GA run	-1000

PSO input parameters	Value
Number of particles in the swarm (n)	40
Number of iterations (k)	40
Inertia weight factor (w)	varies linearly from 0.9 to 0.5
Cognitive and social acceleration factors (c_1 and c_2)	0.5

The GA and PSO input parameters, shown in Table 2, were used on GA and PSO code coupled with the rigorous model obtaining a productivity of 1.84×10^{-4} (kmol)/(m^3s) and 1.95×10^{-4} (kmol)/(m^3s), respectively, as illustrated in Figure 1. The best values of the optimization variables obtained by GA method are (ul = 1.07×10^{-2}, ug = 2.41, ur = 3.0×10^{-3}, Agf = 2.38×10^{-3}, Alf = 2.0×10^{-3}, Blf = 3.81×10^{-2}, Tf = 620.66, Trf = 564.48) and by PSO method are (ul = 1.12×10^{-2}, ug = 2.52, ur = 3.0×10^{-3}, Agf = 2.39×10^{-3}, Alf = 1.39×10^{-3}, Blf = 3.83×10^{-2}, Tf = 621.00, Trf = 573.32).

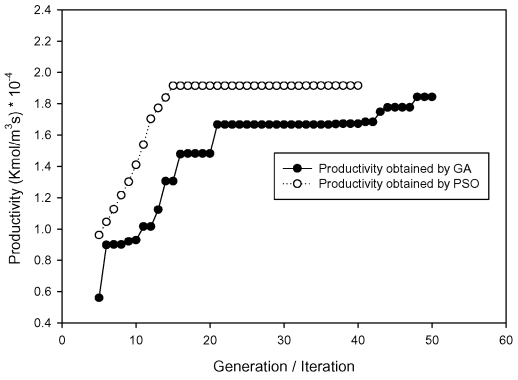

Figure 1 - Comparison of Productivity obtained by PSO and GA methods.

5. Comparison of the Optimisation Performance of PSO and GAs

Comparing the objective function value , Figure 1 showed that with 40 iterations PSO method already reached a better result of the objective function value than GA method that demanded 50 generations. Concerning to the optimal solution found by each method, it can be observed that the values for each input variable are pretty much the same, except for feed coolant temperature indicating the great impact temperature exerts on process productivity. Considering the facility of implementation of the code, it is observed that the GA code requires the set-up of arbitrary precision in decision variables, coding and decoding of the parameters and set-up of parameters that can be done by trial and error method or by factorial design, for instance, while PSO code does not require such steps, which make PSO easier to manipulate than GA. The computational time demanded by PSO and GA are similar, around three minutes, indicating both as appropriate method for real-time implementation.

6. Conclusions

A comparison of the optimisation performance of GA and PSO methods applied to a three-phase catalytic reactor that produces 2-methyl-cyclohexanol was presented. The results of simulations of both methods showed that the PSO objective function value was better than GA objective function value. It was presented that the manipulation of PSO code is easier than GA code. The computational time required by PSO and GA was equivalent, around three minutes, which makes both suitable approaches for real-time optimisation of the three-phase reactor considered in this paper.

Acknowledgements

The authors thank FAPESP, São Paulo, Brazil, for the financial support.

References

Carroll, D., 2008, Carroll's FORTRAN Genetic Algorithm Driver. In http://cuaerospace.com/carroll/ga.html accessed in November 16th , 2008.

Deb, K., 2000, An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics Engineering*, 186, 311-338.

Panda, S.; Padhy, N. P., 2008, Comparison of particle swarm optimisation and genetic algorithm for FACTS-based controller design. Applied Soft Computing, 8, 1418–1427.

Rezende, M.C.A.F., Costa, C.B.B., Costa, A.C., Wolf Maciel, M.R., Maciel Filho, R., 2008, Optimisation of a Large Scale Industrial Reactor by Genetic Algorithms. *Chemical Engineering Science*, v. 63, p. 330-341.

Vasco de Toledo, E.C., Santana, P.L., Wolf Maciel, M.R., Maciel Filho, R. 2001, Dynamic Modelling of a three-phase catalytic slurry reactor. *Chem. Eng. Sc*. 2001, 56, 6055-6061.

Zhang, J.; Xie, L.; Wang, S., 2006, Particle swarm for the dynamic optimisation of biochemical processes. 16th European Symposium on Computer Aided Process Engineering. 2006.