



Process Synthesis and Design using Modified Differential Evolution (MDE)

Rakesh Angira^a and B. V. Babu^{b*}

Department of Chemical Engineering,
Birla Institute of Technology & Science,
Pilani – 333031, India

^a Email: angira@bits-pilani.ac.in; ^b Email: bvbabu@bits-pilani.ac.in

Keywords: Process Synthesis, mixed integer nonlinear programming, and Differential Evolution.

ABSTRACT

A large number of process synthesis and design problems in chemical engineering can be modeled as mixed integer nonlinear programming (MINLP) problems. They involve continuous (floating point) and integer variables. A common feature of this class of mathematical problem is the potential existence of non-convexities due to the particular form of the objective function and/or the set of constraints. Due to their combinatorial nature, these problems are considered to be difficult. In the present study, a novel modified differential evolution (Angira & Babu, 2005) is used for solving process synthesis and design problems. To illustrate the applicability and efficiency of modified differential evolution (MDE), two test problems on process synthesis and design have been solved. These problems arise from the area of chemical engineering, and represent difficult non-convex optimization problems, with continuous and discrete variables. The performance of MDE is compared with that of Genetic Algorithm, Evolution Strategy, and MINLP-Simplex Simulated Annealing (M-SIMPISA).

INTRODUCTION

Process synthesis can be defined as the selection, arrangement, and operation of processing units so as to create an optimal scheme. In other words, it is an act of determining the optimal interconnection of processing units as well as the optimal type and design of units within a process system. The interconnection of processing units is called the structure of the process system. When the performance of the system is specified, the structure of the system and the performance of the processing units are not determined uniquely. This task is combinatorial and open-ended in nature and has received a great deal of attention over the past twenty-five years (Nishida et al., 1981).

The use of mathematical programming techniques for process synthesis has received considerable attention over the last two decades. For example, nonlinear programming (NLP) technique for heat exchanger networks (Floudas et al, 1986), and mixed integer nonlinear programming (MINLP) models for structural flowsheet optimization (Kocis & Grossmann, 1987, 1988, 1989; Floudas et al., 1989) to name a few. The major reason for this increased interest lies in the fact that mathematical programming techniques provide

a systematic framework for process synthesis. Also, there has been substantial progress in methods and software for solving optimization problems, development of powerful modeling languages (General Algebraic Modeling System, GAMS), and technological advances in computing.

In order to formulate the synthesis problem as a mathematical programming problem, a superstructure is postulated which includes many alternate designs from which the optimal process will be selected. Once the superstructure is specified, the next task is to determine the optimal process flow sheet through structural and parameter optimization of the superstructure (which requires the solution of a mixed integer optimization problem). In early 1980s, most of the process synthesis and design problems have been formulated as Mixed-Integer Linear Programming (MILP) problems. Although these formulations have proved to be quite powerful, they have the limitation that nonlinearities in the process equations cannot be treated explicitly and approximated through the discretization. The need for the explicit handling of the nonlinearities in the synthesis problem motivates the use of Mixed-Integer Nonlinear Programming (MINLP). MINLP problems, however, are much more difficult to solve than MILP problems for which Branch and Bound methods perform reasonably well.

The optimization of mixed integer nonlinear programming problems constitutes an active area of research. Gradient optimization techniques have only been able to tackle special formulations, where continuity or convexity had to be imposed, or by exploiting special mathematical structures. Stochastic algorithms, also known as adaptive random search methods, have tackled MINLP problems, mostly in the area of chemical engineering (Salcedo, 1992). Cardoso et al., (1997) compared the performance of the M-SIMPISA with adaptive random search method (MSGA by Salcedo, 1992). They concluded that for small-scale problems and with penalizing scheme (M-SIMPISA-pen) its performance is comparable to MSGA algorithm, however, for large scale and /or ill conditioned problems, the M-SIMPISA performed better. Costa & Oliviera (2001) used GA & Evolution Strategy (ES) and compared the results with M-SIMPISA and M-SIMPISA-pen algorithm. Evolution Strategies emerged as the best algorithms in most of the problems studied. However, ES exhibited difficulties in highly constrained problems but in general, they are found most efficient in terms of function evaluations. Also, all the algorithms (GA, ES, and M-SIMPISA) are found to have great difficulties with multi-product batch plant problem (Grossmann & Sargent, 1979), which is highly constrained; the global optimum corresponds to a point where a very small variation in any of the continuous variables produces infeasibility. Babu & Angira (2002) used DE for solving MINLP problems. DE was found to be computationally efficient.

A large number of process synthesis, design and control problems in chemical engineering can be modeled as mixed integer nonlinear programming problems (Grossmann & Sargent, 1979; Kocis & Grossmann, 1987, 1988, 1989; Floudas et al., 1989; Salcedo, 1992 etc.). They involve continuous (floating point) and integer variables. A common feature of this class of mathematical problem is the potential existence of non-convexities due to the particular form of the objective function and/or the set of constraints. Due to their combinatorial nature, these problems are considered to be difficult.

In the present study, a novel modified differential evolution (Angira & Babu, 2005) is used for solving process synthesis and design problems. To illustrate the applicability and

efficiency of modified differential evolution (MDE), two test problems on process synthesis and design have been solved. These problems arise from the area of chemical engineering, and represent difficult non-convex optimization problems, with continuous and discrete variables. The performance of MDE is compared with that of GA, ES, and M-SIMPSA.

MODIFIED DIFFERENTIAL EVOLUTION (MDE)

The principle of modified DE is same as DE. The major difference between DE and MDE is that MDE maintains only one array as is evident from both the figures. The array is updated as and when a better solution is found. Also, these newly found better solutions can take part in mutation and crossover operation in the current generation itself as opposed to DE (where another array is maintained and these better solutions take part in mutation and crossover operations in next generation). Updating the single array continuously enhances the convergence speed leading to less function evaluations as compared to DE. However, DE maintains two arrays consuming extra memory and CPU-time (more function evaluations). This modification enables the algorithm to get a better trade-off between the convergence rate and the robustness.

Such an improvement can be advantageous in many real-world problems where the evaluation of a candidate solution is a computationally expensive operation and consequently finding the global optimum or a good sub-optimal solution with the original differential evolution algorithm is too time-consuming, or even impossible within the time available. The pseudo code of MDE used in the present study is given below:

Let P a population of size NP ,

and x_i^j the j^{th} individual of dimension D in population P ,

and CR denotes the crossover probability

input $D, NP \geq 4; F \in (0, 1+); CR \in [0, 1]$, and initial

bounds: $lower(x_i); upper(x_i); i = 1, \dots, D$

initialize $P = \{ x_i^1, \dots, x_i^{NP} \}$ as

For each individual $j \in P$

$x_i^j = lower(x_i) + rand_i [0, 1] \times (upper(x_i) - lower(x_i)); i = 1, \dots, D$

end For each

Evaluate P

while the stopping criterion is not satisfied **do**

forall $j \leq NP$

Randomly select $r_1, r_2, r_3 \in (1, \dots, NP)$,

$j \neq r_1 \neq r_2 \neq r_3$

randomly select $i_{rand} \in (1, \dots, D)$

forall $i \leq D$,

$$x_i' = \begin{cases} x_i^{r_3} + F \times (x_i^{r_1} - x_i^{r_2}) & \text{if } (\text{random}[0,1] < CR \wedge i = i_{rand}) \\ x_i^j & \text{otherwise} \end{cases}$$

end forall

$$\text{if } f(\bar{x}') \leq f(\bar{x}^j)$$

$$\text{Then, } \bar{x}^j = \bar{x}'; \quad f(\bar{x}^j) = f(\bar{x}')$$

end forall

end while

Print the results.

TEST PROBLEMS

Problem-1: Two-Reactor Problem

Kocis & Grossmann (1989) first solved this problem using outer approximation (OA) algorithm. The objective here is to select one between two candidate reactors (as shown in Fig. 1) in order to minimize the production cost. Also, it has been solved by Cardoso et al., (1997) using M-SIMPISA, and Costa & Oliveira (2001) using GA and ES.

$$\text{Min } f(x, y_1, y_2, v_1, v_2) = 7.5y_1 + 5.5y_2 + 7v_1 + 6v_2 + 5x$$

Subject to

$$y_1 + y_2 = 1$$

$$z_1 = 0.9[1 - \exp(-0.5v_1)] x_1$$

$$z_2 = 0.8[1 - \exp(-0.4v_2)] x_2$$

$$z_1 + z_2 = 10$$

$$x_1 + x_2 = x$$

$$z_1y_1 + z_2y_2 = 10$$

$$v_1 \leq 10y_1$$

$$v_2 \leq 10y_2$$

$$x_1 \leq 20y_1$$

$$x_2 \leq 20y_2$$

$$x_1, x_2, z_1, z_2, v_1, v_2 \geq 0$$

$$y_1, y_2 \in \{0, 1\}$$

The binary variables y_1 and y_2 denote the existence (nonexistence) of reactor 1 and 2 when their value is 1 (0). In the objective function, there are fixed charges for purchasing reactor 1 (7.5) or reactor 2 (5.5), linear terms in v_1 and v_2 (reactor volumes) and the purchase price for raw material x . The two nonlinear equations are the input-output relations for the reactors which define the output flows (z_1 and z_2) in terms of the input flows (x_1 and x_2) and the volumes. The raw material x is split into the reactor input flows x_1 and x_2 ; a total demand of 10 units must be met by the output flows z_1, z_2 . The next four inequalities are logical constraints which insure that if a given reactor does not exist (e.g. $y_1 = 0$), then the corresponding volume and feed stream are zero. The last constraint requires that either reactor 1 or 2 be selected. The suboptimal solution corresponding to $(y_1, y_2) = (0, 1)$ has an objective function value of 107.376 at $(x_1, x_2) = (0.0, 15.0)$ and $(v_1, v_2) = (0.0, 4.479)$. The global optimum is: $(x, y_1, y_2, v_1, v_2; f) = (13.36227, 1, 0, 3.514237, 0; 99.245209)$.

Problem-1* : Reformulation of Problem-1

This can be reformulated without equality constraints as follows:

$$\text{Min } f(y_1, v_1, v_2) = 7.5y_1 + 5.5(1 - y_1) + 7v_1 + 6v_2$$

$$+ 50 \frac{1 - y_1}{0.8[1 - \exp(-0.4v_2)]} + 50 \frac{y_1}{0.9[1 - \exp(-0.5v_1)]}$$

Subject to

$$0.9[1 - \exp(-0.5v_1)] - 2y_1 \leq 0$$

$$0.8[1 - \exp(-0.4v_2)] - 2(1 - y_1) \leq 0$$

$$v_1 \leq 10y_1$$

$$v_2 \leq 10(1 - y_1)$$

$$v_1, v_2 \geq 0$$

$$y_1 \in \{0, 1\}$$

The global optimum is same as in Problem-1.

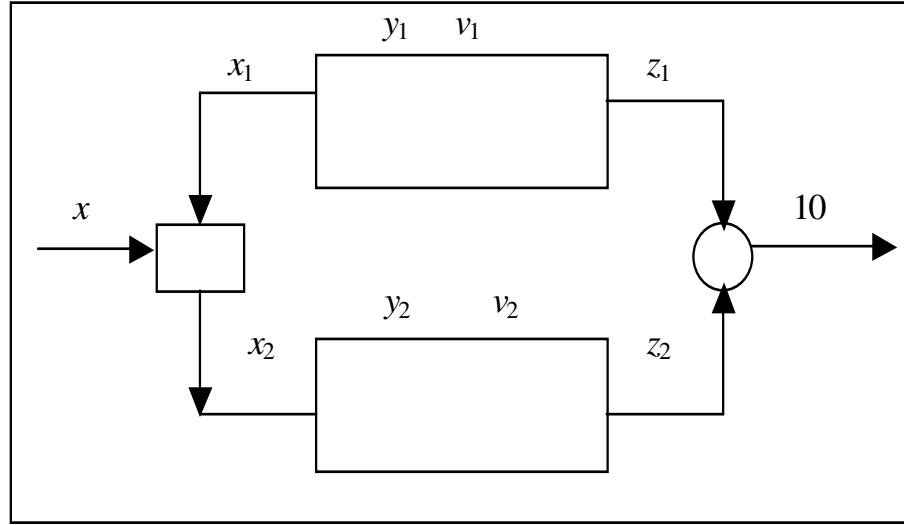


Fig. 1: Superstructure for two-reactor problem

Problem-2: Multi-Product Batch Plant

This is a multi-product batch plant problem with M serial processing stages, where fixed amounts Q_i from N products must be produced. Many researchers studied this problem (Grossmann & Sargent (1979) using branch & bound method, Kocis & Grossmann (1988) using OA, Salcedo (1992) using MSGA, Cardoso (1997) using M-SIMPSA, and Costa & Oliviera (2001) using GA and ES).

$$\text{Min } f = \sum_{j=1}^M \alpha_j N_j V_j^\beta$$

Subject to

$$\sum_{i=1}^N \frac{Q_i T_{Li}}{B_i} \leq H; \quad V_j \geq S_{ij} B_i; \quad N_j T_{Li} \geq t_{ij}; \quad 1 \leq N_j \leq N_j^u; \quad V_j^l \leq V_j \leq V_j^u; \quad T_{Li}^l \leq T_{Li} \leq T_{Li}^u;$$

$B_j^l \leq B_j \leq B_j^u$; where, for the specific problem considered, $M = 3$, $N = 2$, $H = 6000$, $\alpha_j = 250$, $\beta_j = 0.6$, $N_j^u = 3$, $V_j^l = 250$ and $V_j^u = 2500$. The values of T_{Li}^l , T_{Li}^u , B_j^l and B_j^u are given by:

$$T_{Li}^l = \max t_{ij}/N_j^u; \quad T_{Li}^u = \max t_{ij}; \quad B_j^l = Q_i * T_{Li}/H; \quad B_j^u = \min (Q_i, \min_j V_j^u/S_{ij}).$$

The values of S_{ij} and t_{ij} [$i = 1$ to 2 (rows); and $j = 1$ to 3 (columns)] are given in Table 1. The global optimum is: $(N_1, N_2, N_3, V_1, V_2, V_3, B_1, B_2, T_1, T_2; f) = (1, 1, 1, 480, 720, 960, 240, 120, 20, 16; 38499.8)$.

Table 1: Values of S_{ij} and t_{ij} of Problem-2

S_{ij}			t_{ij}		
2	3	4	8	20	8
4	6	3	16	4	4

RESULTS AND DISCUSSION

The two test problems described above not only involves the continuous (floating point) variables but also the integer and/or binary (or discrete) variables. In the present study, integer and binary variables are handled as follows:

Integer Variables

Original DE algorithm is only capable of handling continuous variables. Extending it to optimize integer variables, however, is very easy and requires only couple of simple modifications (Corne et al., 1999). First, integer values should be used to evaluate the objective function, even though DE itself still works internally with continuous floating-point values. Therefore:

$$y_i = \begin{cases} x_i & \text{for continuous variables} \\ INT(x_i) & \text{for integer variables} \end{cases}$$

$INT()$ is a function for converting a real value to an integer value by truncation. Additionally, truncation is performed here for evaluating trial vectors and for handling boundary constraints. Truncated values are not assigned elsewhere. Hence, DE works with a population of continuous variables regardless of the corresponding object variable type which is also essential for maintaining diversity of the population and the robustness of the algorithm.

Binary or Discrete Variables

In the present study, integer variables are handled in the same way as described above however, a different procedure is used to handle binary (discrete) variable. In this procedure, for the function evaluation, binary variables are handled as follows:

$$y_i = \begin{cases} 0 & \text{if } x_i \leq 0.5 \\ 1 & \text{otherwise} \end{cases}$$

Where x_i is a continuous variable $0 \leq x_i \leq 1$. However boundary constraint is handled in the same way as for continuous variables. The only difference is that lower and upper bound are set to zero and one respectively.

For each problem ten executions are carried out with different seed values. Number of function evaluations (NFE), and Number of runs converged to global optimum (NRC) in the subsequent table and figure are the mean values of the ten experiments. The stopping criteria adopted for MDE is to terminate the search process when one of the following conditions is satisfied: (i) the maximum number of generations is reached (assumed 5000 generations for Problem-1* & 10000 generations for Problem-2). (ii) $|f_{\max}^k - f_{\min}^k| < 10^{-5}$ where f is the value of objective function for k -th generation.

Table 2 shows the results obtained using MDE and its comparison with GA, ES, MSIMPSA and MSIMPSA-pen in terms of number of function evaluations. It is clear from Table 2 that NFE is least for MDE among the all methods for the two problems considered in the present study. It is about 92% and 98.16% less than that in GA for the

two problems. It is important to note that ES is found to converge to a non-optimal solution, whereas MDE is able to locate the global optimum for the problems solved in the present study. M-SIMPASA-pen is found to be costlier than GA in terms of computational efforts.

Table 2: Comparison of *NFE* for MDE, GA, ES, M-SIMPASA, and M-SIMPASA-pen

Problem No.	GA	M-SIMPASA	M-SIMPASA-pen	ES	MDE
1*	22489	14738	42295	*	1797
2	225176	#	257536	*	40550

Execution halted

* Converged to a non-optimal solution

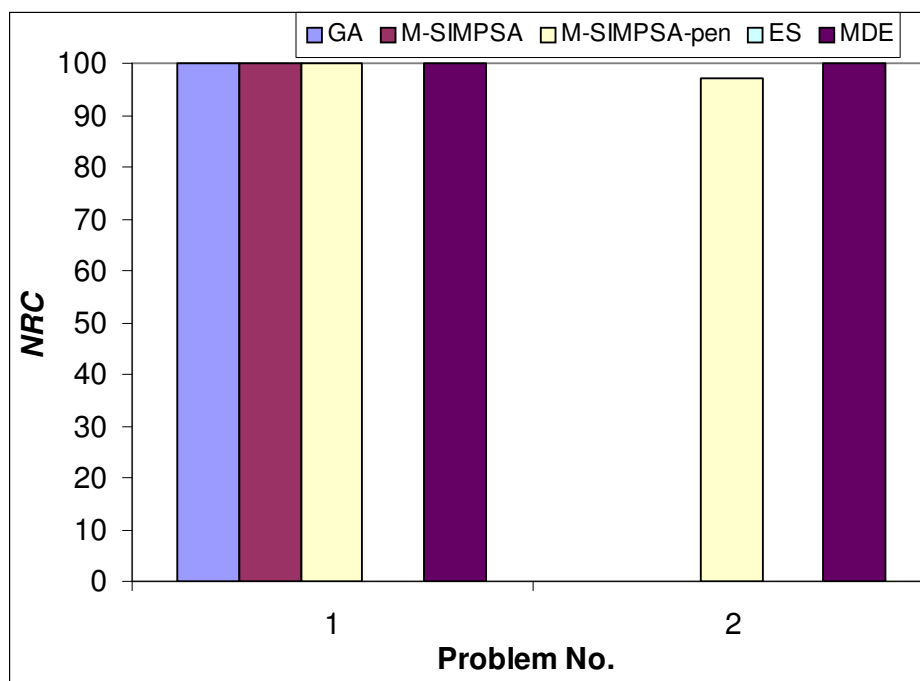


Fig. 2: Comparison of *NRC* for various methods

Figure 2 shows the comparison of *NRC* for various methods. It is evident from Fig. 2 that for Problem-1, all the algorithms except ES are able to locate the global optimum. For second problem, GA, M-SIMPASA, and ES are failed to find the global optimum while M-SIMPASA-pen and MDE are able to locate the global optimum. It is to be noted that only M-SIMPASA-pen and MDE are found to perform better than the other algorithms (GA, ES, M-SIMPASA). However, M-SIMPASA-pen is costlier than MDE in terms of computational efforts. Also, the *NRC* is 97% using M-SIMPASA-pen as compared to 100% using MDE.

CONCLUSIONS

Two test problems on process synthesis and design from chemical engineering are solved using MDE in the present work. The performance of MDE is compared with various algorithms (viz. GA, M-SIMPASA, M-SIMPASA-pen, and ES) in terms of *NFE* and *NRC*. It is found that *NFE* is the least and *NRC* is highest in MDE as compared to GA & M-SIMPASA-pen. The *NRC* in M-SIMPASA-pen is better than that of GA, M-SIMPASA, and ES. The ES and MDE are comparable in terms of *NFE* (rather ES perform slightly better

than MDE) but MDE is found to be robust than ES being higher *NRC* for Problem-1*and 2 (Table 2). The performance of MDE is found to be the best among the methods compared for the problems studied. Hence, MDE algorithm seems to be a better approach for solving process synthesis and design problems

REFERENCES

Angira, R. & Babu, B. V., 2005, Optimization of Non-linear chemical processes using Modified Differential Evolution (MDE), Accepted and to be presented at 2nd Indian International Conference on Artificial Intelligence (IICAI-05), Pune, India, December 20-22.

Babu, B. V. & Angira, R., 2002, A Differential Evolution Approach for Global Optimization of MINLP Problems, *Proceedings of 4th Asia Pacific Conference on Simulated Evolution and Learning (SEAL-2002)*, Singapore, November 18-22, Vol. 2, 880-884.

Cardoso, M. F.; Salcedo, R. L.; Feyer de Azevedo, S.; Barbosa, D., 1997, A simulated annealing approach to the solution of MINLP problems, *Comput.Chem.Engng*, 21, 1349-1364.

Corne, D.; Dorigo, M.; Glover, F., 1999. *New Ideas in optimization*. McGraw-Hill, Berkshire, England (U. K.).

Costa, L. & Oliviera, P., 2001, Evolutionary algorithms approach to the solution of mixed integer no-linear programming problems, *Comput.Chem.Engng*, 25, 257-266.

Floudas, C. A.; Aggarwal, A.; Ciric, A. R., 1989, Global optimum search for nonconvex NLP and MINLP problems, *Comput.Chem.Engng*, 13, 1117 – 1132.

Floudas, C. A.; Ciric, A. R.; Grossmann, I. E., 1986, Automatic synthesis of optimum heat exchanger network configurations, *AIChE Journal*, 32, 276-290.

Grossmann, I. E. & Sargent, R. W. H., 1979, Optimal design of multipurpose chemical plants, *Ind. Eng. Chem. Process design development*, 31, 262.

Kocis, G. R. & Grossmann, I. E., 1987, Relaxation strategy for the structural optimization of process flow sheets, *Ind. Eng. Chem. Res.*, 26, 1869-1880.

Kocis, G. R. & Grossmann, I. E., 1988, Global optimization of nonconvex mixed-integer nonlinear programming problems in process synthesis, *Ind. Eng. Chem. Res.*, 27, 1407-1421.

Kocis, G. R. & Grossmann, I. E., 1989, A modeling and decomposition strategy for the MINLP optimization of process flowsheets, *Comput.Chem.Engng*, 13, 797-819.

Nishida, N.; Stephanopoulos, G.; Westerberg, A. W., 1981, Journal Review: Process synthesis, *AIChE Journal*, 27, 321.

Salcedo, R. L., 1992, Solving Nonconvex Nonlinear Programming Problems with Adaptive Random Search, *Ind. Eng. Chem. Res.*, 31, 262.