Optimization of Non-Linear Chemical Processes using Modified Differential Evolution

Rakesh Angira and B.V. Babu

Department of Chemical Engineering, Birla Institute of Technology & Science Pilani (Rajasthan), India-333 031
{angira, bvbabu}@bits-pilani.ac.in

Abstract. Differential Evolution (DE) is an evolutionary optimization technique that is exceptionally simple, fast, and robust at numerical optimization. However, the convergence rate of DE in optimizing a computationally expensive objective function still does not meet our requirements, and an attempt to speed up DE is considered necessary. This paper introduces a Modified Differential Evolution (MDE) that enhances the convergence rate without compromising the robustness. MDE algorithm utilizes only one set of population array as against two sets in original DE at any given generation. This modification improves the convergence rate of DE and at the same time maintains the robustness. The MDE is applied to two benchmark test functions followed by non-linear chemical processes. The simulation results show empirical evidences on the efficiency and effectiveness of the proposed MDE.

1 Introduction

The field of search and optimization has changed over the last few years by the introduction of a number of non-classical, unorthodox and stochastic search & optimization algorithms. In recent years, Evolutionary Algorithms (EAs) are gaining popularity for the solution of non-linear multimodal problems encountered in many engineering disciplines.

Most of the traditional optimization algorithms based on gradient methods have the possibility of getting trapped at local optimum depending upon the degree of non-linearity and initial guess. Unfortunately, none of the traditional algorithms are guaranteed to find the global optimal solution, but population based algorithms are found to have a better global perspective than the traditional methods [1]. Recently, Onwubolu and Babu [1] compiled new techniques and their applications to various disciplines of engineering and management.

Previous studies [2, 3, 4, 5, 6, 7, 8, 9] have shown that DE is an efficient, effective and robust evolutionary optimization method. Still DE takes large computational time for optimizing the computationally expensive objective functions. And therefore, an attempt to speed up DE is considered necessary. In this paper, Modified Differential Evolution (MDE), an evolutionary optimization technique, is proposed and applied to two benchmark test functions followed by non-linear chemical processes.
2 DE in Brief

Differential Evolution (DE), a recent optimization technique, is an exceptionally simple evolution strategy, which is significantly faster & robust at numerical optimization and is more likely to find a function’s true global optimum [10]. Simple GA [11] uses a binary coding for representing problem parameters whereas DE uses real coding of floating point numbers. Among the DE’s advantages are its simple structure, ease of use, speed and robustness. The details of DE algorithm and pseudo code are available in literature [5, 12].

Original DE dealt with a single strategy [10]. Later on ten different strategies have been suggested by Price and Storn [12]. A set of control parameters that works out to be the best for a given problem may not work well when applied for a different problem. The best value of control parameters to be adopted for each problem is to be determined separately by trial & error. Similarly, the strategy that works out to be the best for a given problem may not be effective when applied to some other problem.

3 Improvements on DE

When using any population based search algorithm in general and DE in particular to optimize a function, an acceptable trade-off between convergence (with reference to locating optimum) and robustness (with reference to not missing the global optima) must generally be determined. Convergence implies a fast convergence although it may be to a local optimum. On the other hand, robustness guarantees a high probability of obtaining the global optimum. A few attempts have already been made to achieve this trade-off [4, 13, 14, 15].

Chiou and Wang [4] embedded accelerated phase and migration phase into the original algorithm of DE. These two phases are used to improve the convergence speed without decreasing the diversity among individuals. Also, several alternate methods are compared. Babu and Angira [13] proposed a variation of mutation and crossover scheme and investigated its effectiveness by applying to liquid extraction problem. Tasoulis et. al. [14], explored how Differential Evolution can be parallelized in a virtual parallel environment so as to improve both the speed and the performance of the method. Bergeya and Ragsdaleb [15] proposed a DE with greedy random strategy for genetic recombination. They found that modified algorithm has higher convergence velocity than original DE still maintaining the robustness. In this paper an attempt has been made to increase the convergence speed of DE without compromising with the robustness. A modified DE is proposed in the present work to achieve this trade-off.

4 Modified Differential Evolution

The principle of modified DE is same as DE. The major difference between DE and MDE is that MDE maintains only one array of population. The array is updated as
and when better solution is found. Also, these 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 better solutions can 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 evaluation). 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 DE algorithm is too time consuming, or even impossible within the time available. The pseudo code of the proposed MDE is given in Fig.1.

**Fig. 1.** Pseudo code of MDE
5 Test Functions

The reliability and efficiency of DE and MDE are tested and compared for two multimodal functions [16]. The selected test functions are briefly described below and details of the global minimum are summarized in Table-1.

Test Function-1. The objective is to minimize the function (GP2).

\[ f = [1 + (A1*A2)][30 + (A3*A4)] \]

Where \( A1 = (x_1 + x_2 +1)^2 \); \( A3 = (2x_1 - 3x_2^2) \);
\( A2 = (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2) \);
\( A4 = (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2) \);
Subject to \(-2\leq x_1, x_2 \leq 2.\)

Test Function-2. The objective is to minimize the function (ES2).

\[ f = -\cos(x_1)\cos(x_2)\exp\left[-((x_1-\pi)^2 + (x_2-\pi)^2)\right] \]
Subject to \(-100 \leq x_1, x_2 \leq 100.\)

Table 1. Details of global minimum

<table>
<thead>
<tr>
<th>Function</th>
<th>( D )</th>
<th>Global minimum</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP2</td>
<td>2</td>
<td>( 3 ) at ( x = {0, -1} )</td>
<td>Four local minima</td>
</tr>
<tr>
<td>ES2</td>
<td>2</td>
<td>(-1 ) at ( x = {\pi, \pi} )</td>
<td>Several local minima</td>
</tr>
</tbody>
</table>

6 Selected Nonlinear Chemical Processes

The optimization of non-linear constraint problems is relevant to chemical engineering practice [17]. Non-linearities are introduced by process equipment design relations, by equilibrium relations and by combined heat and mass balances. There are many chemical processes which are highly nonlinear and complex with reference to optimal operating conditions with many equality and inequality constraints. In this paper the following processes are considered for applying DE and MDE: (1) optimal operation of alkylation unit, and (2) dynamic optimization of a batch reactor.

6.1 Optimal Operation of Alkylation Unit

Alkylation process is common in the petroleum industry. A simplified process flow diagram of an alkylation process is shown in Fig. 2. The process model was described and solved in literature [18] using successive linear programming. The process model
seeks to determine the optimum set of operating conditions for the process, based on a mathematical model, which allowed maximization of profit. Bracken and McCormick [19] formulated the problem as a direct nonlinear programming model with mixed nonlinear inequality and equality constraints and a nonlinear profit function to be maximized. They used Sequential Unconstrained Minimization Technique (SUMT) for solving the same. Later, Dembo [20] transformed the NLP problem with ten variables which Bracken and McCormick [19] derived, into a problem with seven variables. All equality constraints are eliminated and the problem has been formulated as a signomial optimization problem. This problem involves seven variables subject to twelve nonlinear and two linear inequality constraints. Edger and Himmelblau [21] used sequential quadratic programming to solve the problem as formulated by Bracken and McCormick [19]. Maranas and Floudas [22] used generalized geometric programming to solve the seven variables problem as formulated by Dembo [20]. Adjiman et al. [23] used αBB algorithm (for general twice–differentiable constraint NLPs) for solving this problem.

As shown in Fig. 2, an olefin feed (100% butane), a pure isobutane recycle and a 100% isobutane make-up stream are introduced in a reactor together with an acid catalyst. The reactor product stream is then passed through a fractionator where the isobutane and the alkylate product are separated. The spent acid is also removed from the reactor. The variables are defined as shown in Table-2 along with the upper and lower bounds on each variable. The bounds represent economic, physical and performance constraints. In the present study, the problem formulation is same as that of [22, 23]. The problem is briefly discussed below.

![Fig. 2. Simplified Alkylation Process Flow sheet](image-url)
Table 2. Variables and their bounds

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Variable</th>
<th>Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>Olefin feed rate (barrels/day)</td>
<td>1500</td>
<td>2000</td>
</tr>
<tr>
<td>$x_2$</td>
<td>Acid addition rate (thousands of pounds/day)</td>
<td>1</td>
<td>120</td>
</tr>
<tr>
<td>$x_3$</td>
<td>Alkylate yield (barrels/day)</td>
<td>3000</td>
<td>3500</td>
</tr>
<tr>
<td>$x_4$</td>
<td>Acid strength (wt. %)</td>
<td>85</td>
<td>93</td>
</tr>
<tr>
<td>$x_5$</td>
<td>Motor octane no.</td>
<td>90</td>
<td>95</td>
</tr>
<tr>
<td>$x_6$</td>
<td>External Isobutane-to-olefin Ratio</td>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>$x_7$</td>
<td>F-4 performance no.</td>
<td>145</td>
<td>162</td>
</tr>
</tbody>
</table>

Profit Function. The objective is to improve the octane number of some olefin feed by reacting it with isobutane in the presence of acid. The product of the reaction is distilled and the un-reacted is recycled back to the reactor. The objective function was defined in terms of alkylate product, or output value minus feed and recycle costs. Operating costs were not reflected in the function. The total profit ($ per day), to be maximized [23], is given as follows:

Max. \[ \text{Profit} = 1.715x_1 + 0.035x_1x_6 + 4.0565x_3 + 10.0x_2 - 0.063x_3x_5 \]

Subject to the following constraints:

\[
\begin{align*}
0.0059553571x_6^2x_1 + 0.88392857x_3 - 0.1175625x_6x_1 - x_1 & \leq 0 \\
1.1088x_1 + 0.1303533x_1x_6 - 0.0066033x_1x_6^2 - x_3 & \leq 0 \\
6.66173269x_6^2 + 172.39878x_5 - 56.596669x_4 - 191.20592x_6 - 10000 & \leq 0 \\
1.08702x_6 + 0.32175x_4 - 0.03762x_6^2 - x_5 + 56.85075 & \leq 0 \\
0.006198x_7x_4x_3 + 2462.3121x_2 - 25.125634x_2x_4 - x_3x_4 & \leq 0 \\
161.18996x_3x_4 + 5000.0x_2x_4 - 489510.0x_2 - x_3x_4x_7 & \leq 0 \\
0.33x_7 - x_5 + 44.333333 & \leq 0 \\
0.022556x_5 - 0.007595x_7 - 1.0 & \leq 0 \\
0.0006lx_3 - 0.0005x_1 - 1.0 & \leq 0 \\
0.819672x_1 - x_3 + 0.819672 & \leq 0 \\
24500.0x_2 - 250.0x_2x_4 - x_3x_4 & \leq 0 \\
1020.4082x_4x_2 + 1.2244898x_3x_4 - 100000x_2 & \leq 0 \\
6.25x_1x_6 + 6.25x_1 - 7.625x_3 - 100000 & \leq 0 \\
1.22x_3 - x_6x_1 - x_1 + 1.0 & \leq 0
\end{align*}
\]
The maximum profit as reported in [23] is: $1772.77 per day, and the optimal variable values are $x_1 = 1698.18$, $x_2 = 53.66$, $x_3 = 3031.3$, $x_4 = 90.11$, $x_5 = 95.0$, $x_6 = 10.50$, $x_7 = 153.53$.

6.2 Dynamic Optimization of a Batch Reactor

In this problem, we consider the consecutive reaction: $A \xrightarrow{k_1} B \xrightarrow{k_2} C$ in a batch reactor [24]. $C_A$ and $C_B$ are the concentration of component $A$ and $B$ respectively and $T$ is the temperature of reactor. The first reaction is of second order and the second reaction is of first order and $k_1$ and $k_2$ are the rate constants respectively for the first and second reaction. The objective is to obtain the optimal reactor temperature progression, which maximizes the intermediate product $B$ for a fixed batch time.

The dynamics are given by the following design equations:

$$\frac{dC_A}{dt} = -k_1 C_A^2$$
$$\frac{dC_B}{dt} = k_1 C_A^2 - k_2 C_B$$

Where $k_1 = 4000 \exp\left(-\frac{2500}{T}\right)$

$$k_2 = 620000 \exp\left(-\frac{5000}{T}\right)$$

and $298 \leq T \leq 398$

The initial conditions are $C_A(0) = 1.0$, and $C_B(0) = 0.0$. And the objective function is $J = \text{Maximize } C_B$. We need to find out optimal temperature profile, which gives maximum intermediate product concentration. This problem has been solved in [25] using piecewise constant controls. Dadebo and Mcauley [26] used dynamic programming for solving this problem and reported results for different number of stages.

In this problem our parameters are temperatures at different time intervals. With these temperatures we need to find out the optimal final concentration of $B$ (by solving the above model equations along with DE and MDE).

7 Results and Discussion

In section 4 and 5, we have discussed the two test functions and selected chemical engineering problems. Most of the engineering optimization problems are constrained and there are many methods to handle it. The following subsection discusses the handling of constraint in the present study followed by results and discussion of test functions and selected optimization problems from chemical engineering.
7.1 Constraint Handling

The difficulty of using EAs in the constrained optimization is that the evolutionary operators used to manipulate the individuals of the population often produce solution which are unfeasible. This unfeasibility may be due to bound violations of decision variable (whether upper or lower) or due to violation of constraints. In the present work, bound violation is repaired by generating the parameter randomly between given lower and upper bounds. However, the penalty function method is used to handle the constraints. This is one of the most popular techniques in EAs to handle constraints. The technique transforms the constrained problem into an unconstrained problem by penalizing unfeasible solutions. In addition, the penalty function methods are easy to implement and considered efficient.

7.2 Test Functions

Table 3 shows the results obtained using DE and MDE. Both the methods are coded in C++ using BorlandC++ on PC with Pentium III, 500 MHz/128 MB RAM. The strategy used is DE/rand/1/bin and the key parameters used are NP = 10D, CR = 0.5, F = 0.8. The results are compared on the basis of average CPU-time and success rate (NRC) in locating the global optimum for the given tolerance, in an overall ten executions implemented.

Table 3. Results of DE and MDE for test functions

<table>
<thead>
<tr>
<th>S. No.</th>
<th>(NFE) (CPU-time)</th>
<th>Percentage Time saving</th>
<th>(NRC) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DE</td>
<td>MDE</td>
<td></td>
</tr>
<tr>
<td>ES2</td>
<td>3052 (0.094)</td>
<td>2512 (0.083)</td>
<td>11.7</td>
</tr>
<tr>
<td>GP2</td>
<td>1222 (0.020)</td>
<td>1024 (0.016)</td>
<td>20.0</td>
</tr>
</tbody>
</table>

The termination criterion used is \(|\text{OF}_{\text{cal}} - \text{OF}_{\text{Anal}}| \leq 1 \times 10^{-6}\) (where \(\text{OF}_{\text{cal}}\) is the objective function value at the best point found in each successful run and \(\text{OF}_{\text{Anal}}\) is the known global minimum) for both DE & MDE for comparison purposes. It is clear that MDE takes less CPU-time and hence number of function evaluations. Also, the success rate is 100% for the two test functions using DE and MDE. There is a saving in CPU-time of about 11 to 20% in two test functions using MDE.

Fig. 3 shows the convergence history of DE and MDE for the test function named ES2. Error is the difference between average cost of population in a generation and cost or objective function value corresponding to known global optimum. It is clear that error reduces to 0.0014 after 52 generations (for MDE) while for DE it reduces to 0.00167 only even after 73 generations. This clearly explains the higher speed of MDE to attain global optimum.
Fig. 3. Convergence History of Test function ES2

### 7.3 Selected Nonlinear Chemical Processes

**Alkylation Process Optimization.** Table-4 presents the comparison of results obtained in earlier studies and, those obtained using DE in present study. These results cannot be compared because of the different platform used. But, there is a variation in value of objective function obtained using different optimization methods. The optimal solution obtained using DE is: \( x_1 = 1698.256922; \ x_2 = 54.274463; \ x_3 = 3031.357313; \ x_4 = 90.190233; \ x_5 = 95.0; \ x_6 = 10.504119; \ x_7 = 153.535355; \) and the value of objective function is 1766.36. This solution satisfies all the inequality constraints to six decimal places (i.e. 0.0000001) while solution reported by [22, 23] violates the first, third and sixth constraints. The values of first, third and sixth constraints are found to be 0.016501, 4.752125, and 1727.867362 respectively instead of zero or less than zero. Hence the better solution (possibly global optimum) is 1766.36 that satisfy all the constraints to at least six decimal places.
Table 4. Comparison of Various Methods

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Methods</th>
<th>Objective function ((f))</th>
<th>CPU-time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SUMT [19]</td>
<td>1769</td>
<td>Not reported</td>
</tr>
<tr>
<td>2</td>
<td>NPSOL [21]</td>
<td>1768.75</td>
<td>Not reported</td>
</tr>
<tr>
<td>3</td>
<td>GGP in GAMS [22]</td>
<td>1773</td>
<td>30</td>
</tr>
<tr>
<td>4</td>
<td>(\alpha)BB algorithm [23]</td>
<td>1772.77</td>
<td>13.62</td>
</tr>
<tr>
<td>5</td>
<td>Differential Evolution (present study)</td>
<td>1766.36</td>
<td>5.77</td>
</tr>
<tr>
<td>6</td>
<td>Modified Differential Evolution (present study)</td>
<td>1766.36</td>
<td>5.43</td>
</tr>
</tbody>
</table>

Table 5. Results\(^5\) of DE and MDE for Alkylation Problem

<table>
<thead>
<tr>
<th>Methods</th>
<th>(NFE)</th>
<th>(NRC\ (%))</th>
<th>CPU-time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DE</td>
<td>114895</td>
<td>100</td>
<td>5.77</td>
</tr>
<tr>
<td>MDE</td>
<td>108103</td>
<td>100</td>
<td>5.43</td>
</tr>
</tbody>
</table>

Table-5 shows the results obtained using DE and MDE and their comparison in terms of the number of objective function evaluations, CPU-time and the percentage of convergencies to the optimum, i.e., success rate. The termination criterion used is an accuracy value of \(1 \times 10^{-6}\). \(NFE\), \(NRC\) and CPU-time in Table-5 represents, respectively the mean number of objective function evaluations over all the 10 experiments (with different seed values), success rate, and the average CPU time per experiment. The key parameters used are \(NP = 10D\), \(CR = 0.8\), \(F = 0.5\). Both DE and MDE are able to locate the global optimum in all the experiments, as \(NRC\) is 100%. MDE takes 5.9% less CPU-time than DE. Results in Table-5 clearly indicate that MDE is faster than DE and takes less CPU-time to locate the global optimum solution.

**Dynamic Optimization of a Batch Reactor.** In this problem, we need to find out optimal temperature profile, which gives maximum intermediate product concentration. This problem has been solved in [25] using piecewise constant controls. They [25] reported a value of 0.61 for the objective function. Dadebo and Mcauley [26] used dynamic programming for solving this problem. They reported results for different number of stages. They [26] reported a yield of 0.610070 for 10 stages which is same as shown in Table-6 for DE and MDE. In the Table-6, CPU-time is average of 10 experiments. It is evident that MDE takes about 13.38% less CPU-time than that of DE. Also, it is to be noted that a considerable time saving (20.3s in the present case) can be achieved in such type of problems as compared to other problems where objective function is not computationally expensive (Table-3).

---

1 CPU-time on HP-730 workstation
2 CPU-time on HP9000/730 using scaled Gerschgorin theorem method
3 CPU-time on PC with Pentium PIII, 500 MHz/128 MB RAM/ 10 Gb HD (strategy DE/rand/1/bin)
4 CPU-time on PC with Pentium PIII, 500 MHz/128 MB RAM/ 10 Gb HD (strategy DE/rand/1/bin)
5 Penalty used is \(1 \times 10^5\)
Table 6. Results of DE and MDE for 10 intervals

<table>
<thead>
<tr>
<th>Methods</th>
<th>Yield</th>
<th>Optimal Initial Temperature $T(0)$</th>
<th>CPU-time(s)</th>
<th>NRC (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DE</td>
<td>0.610079</td>
<td>361.4</td>
<td>151.7</td>
<td>100</td>
</tr>
<tr>
<td>MDE</td>
<td>0.610079</td>
<td>361.4</td>
<td>131.4</td>
<td>100</td>
</tr>
</tbody>
</table>

Fig. 4 shows the optimal temperature profile obtained using DE and MDE for 10 intervals of total time (i.e. for 10 stages). Both the profiles are exactly same. Having obtained the same profile in lesser CPU-time establishes the fact that MDE is able to find the optimal temperature profile faster than DE.

![Optimal Temperature Profile for 10 intervals](image)

**8 Conclusions**

The Modified Differential Evolution (MDE) algorithm has been introduced and compared with Differential Evolution (DE) for global optimization of benchmark test

---

6 On Pentium-4/2.4GHz/256 MB (RAM) using R-K method of 4th order with step size of 0.0001
functions and selected nonlinear chemical processes. The results stated above clearly show the improvement upon the performance characteristics of DE with regard to the number of function evaluations (NFE)/CPU-time required to find the global optimum. The enhancement was accomplished by using single array in MDE as compared to two arrays in DE. The second method i.e. without forcing the bound is found to be better than the forced bound method, as it avoids trapping at local optimal solution if they are present at extreme (upper or lower bound).

In this paper, we focused applying DE & MDE to nonlinear unconstrained and constrained problems encountered in chemical engineering. An interesting area of future study would involve the application of MDE to other numeric domains such as mixed integer problems or other problem domains such as multi-criteria optimization.

References


