Optimization of reactor network design problem using Jumping Gene Adaptation of Differential Evolution

Ashish M. Gujarathi*, S. Purohitb and B. Srikanthc

aPetroleum and Chemical Engineering Department
Sultan Qaboos University, Muscat, PC 123, Sultanate of Oman

bBirla Institute of Technology and Science, Pilani, India.

ashishg@squ.edu.om; ashishgujrathi@gmail.com

Abstract. Detailed working principle of jumping gene adaptation of differential evolution (DE-JGa) is presented. The performance of the DE-JGa algorithm is compared with the performance of differential evolution (DE) and modified DE (MDE) by applying these algorithms on industrial problems. In this study Reactor network design (RND) problem is solved using DE, MDE, and DE-JGa algorithms: These industrial processes are highly nonlinear and complex with reference to optimal operating conditions with many equality and inequality constraints. Extensive computational comparisons have been made for all the chemical engineering problems considered. The results obtained in the present study show that DE-JGa algorithm outperforms the other algorithms (DE and MDE). Several comparisons are made among the algorithms with regard to the number of function evaluations (NFE)/CPU-time required to find the global optimum. The standard deviation and the variance values obtained using DE-JGa, DE and MDE algorithms also show that the DE-JGa algorithm gives consistent set of results for the majority of the test problems and the industrial real world problems.

1. Introduction
Optimization refers to finding the best value of solution among the set of feasible solutions. In case of single objective optimization, there may exist a single global optimum solution and several near global or the local optimal solutions. Thus, the ideal outcome of a single objective optimization problem is a single global solution. The traditional algorithms often start with a single point (guess value) and end up with a single point solution. However, the outcome of gradient based traditional algorithms largely depends on its control parameters such as the step size and the direction of search that are being used. In a complex and non-linear search space, which may involve multiple solutions (such as local optima and a global optimum solution), an inefficient search algorithm may get trapped at local optimal solution. In contrast, evolutionary algorithms which mimic nature’s principle of survival of the fittest start with multiple population points [1-4]. Due to the strong genetic operators, evolutionary algorithms are found to achieve the global optimum in majority of industrial applications both for single objective [5] and multi-objective optimization [6]. Thus it is necessary to develop an optimization algorithm which not only captures and global solutions, but also acquires it at a faster rate.
In the recent years, non-traditional search and optimization methods based on natural phenomena, such as Simulated Annealing (SA), Genetic algorithms (GA), Differential evolution (DE), Self Organizing Migrating Algorithms (SOMA), Particle Swarm Optimization (PSO), Tabu Search (TS), Scatter Search (SS), and Ant Colony Optimization (ACO) have been developed to overcome the problems of traditional optimization methods. These algorithms are stochastic in nature and are comparatively new and are gaining popularity due to certain properties which the deterministic algorithms do not have. Thus these are found to have better global perspective than the traditional methods [4, 7].

1.1. Differential evolution in brief
Differential evolution (DE), an evolutionary algorithm is an extension of genetic algorithm (GA) for solving single objective optimization problems. Simple GA [3] uses a binary coding for representing problem parameters whereas DE uses real coding of floating point numbers. DE 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 [8]. It is found that DE is a novel optimization method which is capable of handling nonlinear, multimodal and non-differentiable optimization problems [5]. Original DE dealt with a single strategy [8]. Later on ten different strategies have been suggested by [9]. 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 [5]. 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, 7, 9, 10]. DE is also extended to solve multi-objective optimization problems. Several variants of multi-objective differential evolution (MODE) algorithm (namely, MODE-II, MODE-III, Hybrid-MODE, Elitist-MODE, Trigonometric mutation-MODE and jumping gene variants of MODE) are available in literature. The following sections give the detailed working principles of differential evolution with jumping gene adaptation algorithm [11-17].

1.2. Differential evolution with jumping gene adaptation
Transposons, also called jumping genes, are sequences of DNA that can move around to different positions within the genome of a single cell, a process called transposition. In the process, they can cause mutations and change the amount of DNA in the genome. Thus, they promote inter and intra chromosome movement [18].

A new jumping gene based differential evolution algorithm is developed which is an improvement to the modified differential evolution [10]. Jumping gene phenomenon has already been applied to evolutionary techniques both binary and real coded like genetic algorithm and simulated annealing for multi-objective optimization [19-21]. It was observed that implementation of jumping gene operation to the original evolutionary algorithm helps in achieving the optimum solution at a faster rate. Though the concept of jumping gene is applied on genetic algorithms and simulated annealing algorithm, to the best of our knowledge this is the first attempt to implement the jumping gene operation in differential evolution algorithm.

According to the experimental observation jumping genes can move around the genome in two ways [18]: Cut and paste, and Copy and paste. Cut and paste means a piece of DNA is cut and pasted somewhere else. The latter means the genes remain at the same location while the message in the DNA is copied into RNA and then copied back into DNA at another place in the genome. Whichever the process of transposition, the jumping genes in one place can affect the genetic behaviour at other places in the genome [18]. Jumping gene may allow searching the entire search space (during the initial generations) and thus it creates more chances of achieving a better convergence and diversity and avoiding convergence to a local maxima/minima.
The jumping gene adaptation is applied immediately after the replacement of target vector by the trial vector (i.e., within the population loop). Following pseudo code gives the working of jumping gene algorithm based on single chromosome. Two kinds of mechanisms are considered, viz. cut and copy as given below.

%% Cut and paste in single chromosome
Step1: Identify ins_pos, gene_pos and gene_length
X_initial is from the current population
Copy X_initial to X_jump
Step2: Check for bounds at positions gene_pos to gene_pos + gene_length-1 and corresponding values at ins_pos to ins_pos+gene_length-1
If the bounds are different then map the elements at position gene_pos to gene_pos+gene_length-1 and copy to X_gene else copy the original X_initial
Step3: Remove elements from gene_pos to gene_pos+gene_length-1 from X_jump
Step4: Copy elements at position 1 to (ins_pos-1) in X_jump to xjg1
Copy Elements at position ins_pos:end to xjg2. Now we have 3 vectors. Con-catenate these vectors to form X_jump_trial
Step5: Thus, X_jump_trial = xjg1 + X_gene + xjg2
Step6: Compare the cost of X_jump_trial with the cost of target vector and replace the target if better point is achieved

%% Copy and paste in single chromosome
Step1: Identify ins_pos, gene_pos and gene_length
X_initial is from the current population
Step2: Copy elements at position gene_pos to gene_pos+gene_length-1 to X_gene
Step3: Change elements from ins_pos to ins_pos+gene_length -1 with values from X_gene and name this new vector as X_jump_trial
No shifting of elements of an array is required
Step4: Compare the cost of X_jump_trial with the cost of target vector and re-place the target if better point is achieved

More variants of cut and paste and copy and paste in double chromosomes are already designed for both single and multi-objective differential evolution and is available in literature [22]. However in this study results obtained for only single chromosome based JG are reported. In our previous study it was observed that the differential evolution with jumping gene adaptations (DE-JGa algorithm) is able to consistently achieve a value of standard deviation close to zero for majority of the test problems [23]. In terms of best and worst solutions (among the finally obtained solutions) DE-JGa algorithm was able to provide best and worst solutions which are closest to the global solutions. The mean values of solutions obtained using DE-JGa algorithm were consistently close to global optimum solutions (exact global solutions for many case studies) and were achieved at a faster rate compared to those obtained using DE and MDE algorithms. Inspiring from the results obtained on test problems, DE-JGa algorithm is used solve industrial real world problems.

2. Problem Formulation

In this study reactor network design (RND) problem is considered for optimization study. RND problem is optimized using variants of differential evolution algorithms namely, DE, MDE, and DE-JGa.

This example, taken from Ryoo and Sahinidis [24], is a reactor network design problem, describing the system shown in Figure 1. It involves the design of a sequence of two CSTR reactors where the consecutive reaction A→B→C takes place. The objective function considered in this study is the maximization of the concentration of product B (i.e., \( x_4 = C_{B2} \)) in the exit stream. This problem is known to have caused difficulties for other global optimization methods. Mathematical, the problem is represented as given through Eqs. 1-2f:
Min. \( f = -x_4 \) (1)

Subject to

\[
\begin{align*}
4x - x_1 + k_2 x_2 x_6 &= 0 \\
x_3 + x_1 + k_3 x_3 x_5 &= 1 \\
x_4 - x_3 + x_1 + k_4 x_4 x_6 &= 0 \\
x_5^{0.5} + x_6^{0.5} &= 4 \\
(0, 0, 0, 0, 10^{-5}, 10^{-5}) &\leq (x_1, x_2, x_3, x_4, x_5, x_6) \leq (1, 1, 1, 16, 16) 
\end{align*}
\]

Where

\[
\begin{align*}
k_1 &= 0.09755988 \\
k_2 &= 0.99k_1 \\
k_3 &= 0.0391908 \\
k_4 &= 0.9k_3 
\end{align*}
\]

The global optimum is: \((x_1, x_2, x_3, x_4, x_5, f) = (0.771462, 0.516997, 0.204234, 0.388812, 3.036504, -0.388812)\).

This example constitute a very difficult test problem as it possesses a local minimum with an objective function value that is very close to that of the global solution. The local solutions are with \( f = -0.375 \) and \( f = -0.3881 \) [5, 10]. Interestingly enough, the two local solutions utilize only one of the two reactors whereas the global solution makes use of both reactors. This problem can be reformulated by eliminating equality constraint as follows (Eqs. 3-5):

\[
\begin{align*}
Max. \quad f &= \frac{k_2 x_6 (1 + k_1 x_4) + k_4 x_4 (1 + k_2 x_6)}{(1 + k_1 x_3) (1 + k_2 x_6) (1 + k_3 x_5) (1 + k_4 x_6)} \\
Subject to \quad x_5^{0.5} + x_6^{0.5} &\leq 4 \\
(10^{-5}, 10^{-5}) &\leq (x_5, x_6) \leq (16, 16) 
\end{align*}
\]
Global optimum is same after reformulation. In the present study, the reformulated problem is solved using DE, MDE and Jumping gene adaptation of DE.

3. Results and Discussion

Table 1 show the results obtained using DE, MDE and DE-JGa for the optimization of reactor network design problem. The results are shown considering average best, worst, mean, median, and standard deviation of the optimum values of 25 runs. The term ‘Error’ considered in this study is the difference between average cost of population and cost or objective function value corresponding to known global optimum in a generation. Though the experimental runs are taken for 500 numbers of generations, the comparison of results is carried out at intermediate numbers of functions evaluations (FES) i.e., FES value of 10000,15000,20000,50000 and 100000 (Corresponding to 50, 75, 100, 250 and 500 generations respectively).

| Algorithm | FES     | Best      | Worst      | Mean       | Median     | Std-Dev   |
|-----------|---------|-----------|------------|------------|------------|-----------|
| DE        | 10000   | 0.388811  | 0.388453   | 0.388789   | 0.388793   | 1.81E-05  |
|           | 15000   | 0.388811  | 0.388804   | 0.388811   | 0.388811   | 2.46E-07  |
|           | 20000   | 0.388811  | 0.388811   | 0.388811   | 0.388811   | 2.88E-09  |
|           | 50000   | 0.388811  | 0.388811   | 0.388811   | 0.388811   | 0         |
|           | 100000  | 0.388811  | 0.388811   | 0.388811   | 0.388811   | 0         |
| MDE       | 10000   | 0.388811  | 0.38723    | 0.38807    | 0.38806    | 4.62E-06  |
|           | 15000   | 0.388811  | 0.388811   | 0.388811   | 0.388811   | 2.92E-08  |
|           | 20000   | 0.388811  | 0.388811   | 0.388811   | 0.388811   | 2E-10     |
|           | 50000   | 0.388811  | 0.388811   | 0.388811   | 0.388811   | 0         |
|           | 100000  | 0.388811  | 0.388811   | 0.388811   | 0.388811   | 0         |
| DE-JGa    | 10000   | 0.38881   | 0.37583    | 0.387298   | 0.387632   | 0.001278  |
|           | 15000   | 0.38881   | 0.38647    | 0.38863    | 0.388671   | 0.000155  |
|           | 20000   | 0.38881   | 0.38857    | 0.388793   | 0.388798   | 1.63E-05  |
|           | 50000   | 0.38881   | 0.38881    | 0.38881    | 0.38881    | 0         |
|           | 100000  | 0.38881   | 0.38881    | 0.38881    | 0.38881    | 0         |

Taking standard deviation to be one of the measures of the effectiveness of any algorithm, it is observed that MDE produces better results than those obtained using DE and DE with jumping gene adaptation. The consistency with which DE and MDE has performed is evident from the fact that standard deviation achieved a value of 0 (zero) or close to zero compared to that obtained using DE combined with jumping gene (DE-JGa) algorithm. For most of the test problems and industrial case studies, the jumping gene strategy of DE algorithm is able to give consistent results at a faster rate as compared to those obtained using differential evolution and modified differential evolution algorithms. Table 1 show the results obtained using DE, MDE and DE-JGa for the optimization of reactor network design problem. The standard deviation values at FES value of 20000 obtained using DE, MDE and DE-JGa algorithms are 2.88x10^-09, 2x10^-10, and 1.63x10^-05 respectively. For reactor network design problem MDE algorithm is able to produce better results that those obtained using DE and MDE and DE-JGa algorithms. The consistency with which all the strategies of DE have performed is evident from the fact that standard deviation achieved a value of 0 (zero).
2 show that error reduces faster in MDE compared to DE and DE-JGa in the first few generations. But after 36 generations, DE-JGa converges to zero faster than MDE. However when given enough opportunity, we see that jumping gene algorithm performs exceptionally well among all the three. Using both DE and MDE algorithms the final value of error converges to $5.67 \times 10^{-7}$ and it converge to the value of zero using DE-JGa algorithm. Thus the global optimum value is obtained using DE-JGa algorithm and a value close to global optimum is obtained using by DE and MDE algorithms in this case study (RND problem).

![Error against generation number for RND problem](image)

**Figure 2.** Error against generation number for RND problem

3.1. Algorithm Specific Study
In this section each of the algorithm is tested for its robustness using various values of control parameters such as (crossover constant) CR and F (scaling factor). The important evaluation parameters such as FES, CPU time and predicted function value are also obtained for each of the problems using strategies of differential evolution algorithm.

3.1.1. Effect of control parameters (CR and F)
It is possible that the output of stochastic algorithms such as DE and its variants may depend on its control parameters (CR and F). Thus an attempt is made to check the effects of those control parameters on the performance of algorithms on the selected test problems. During the study of effect of CR, the value of scaling factor was generated randomly whereas during the study of effect of scaling factor, the value of CR was set to 0.8 (based on our previous experience). Figures 3a-3b shows the effect of CR and F on the objective functions cost (function value). Each point on the figures (Figure 3a-3b) represents the average of 25 experiments.
Figure 3. Effect of (a) CR and (b) F on cost of reactor network design problem

Table 3 shows FES, CPU time and function value for various problems using strategies of DE. In all these problems, MDE is able to locate the global optimum in lesser number of FES than DE and MDE. DE is able to locate the global optimum in lesser number of FES than DE-JGa. The CPU time taken by MDE is much lesser compared to DE and DE-JGa. Both DE and MDE are is able to converge to optimum value faster than DE-JGa. The DE-JGa algorithm converges to optimum for various other industrial problems (which are not reported in this study).

Table 3. Number of function evaluations, CPU time and function value

| Problem | DE   | MDE  | DE-JGa |
|---------|------|------|--------|
| RND     | 14283| 12699| 23488  |
| CPU Time (s) | 3.634823 | 2.979619 | 4.58643 |
| Function Value (Cost) | 0.3888119 | 0.3888119 | 0.3888119 |

4. Conclusions:
A new algorithm, differential evolution with jumping gene adaptations (DE-JGa) is discussed. The performance of the newly developed algorithm is compared with the performance of existing algorithms (DE and MDE) by applying these algorithms to some selected industrial problems. Detailed algorithm specific study is carried out to check the robustness of individual algorithms with respect to its control parameters. Following conclusions are drawn based on the results obtained in the present study.

1. In case of problems which are having less number of decision variables the output of all the algorithms (considered in this is study) is independent of the value of the CR.
2. MDE is able to consistently achieve value of standard deviation close to zero for majority of the test problems considered in this study.
3. DE-JGa algorithm is able to give the objective function value close to optimum over a wide range of CR and F compared to DE and MDE algorithms.
4. In terms of FES and CPU time, MDE algorithm outperforms DE and DE-JGa algorithms

References
1. Goldberg DE 1989 Genetic Algorithms for Search, Optimization, and Machine Learning. (Reading, M.A.: Addison-Wesley)
2. Deb K 2001 *Multi-Objective Optimization using Evolutionary Algorithms* (New York: John Wiley & Sons Limited).
3. Deb K 2005 *Optimization for engineering design algorithms and examples* (Prentice-Hall of India Pvt. Ltd)
4. Babu BV 2004 *Process plant simulation*. (New York: Oxford University Press)
5. Angira R 2006 *Evolutionary Computation for Optimization of Selected Non-linear Chemical Processes. Ph.D. Thesis* (BITS-Pilani, India)
6. Gujarathi A M 2010 *Pareto Optimal Solutions in Process Design Decisions using Evolutionary Multi-objective Optimization. Ph.D. Thesis* (BITS-Pilani, India)
7. Corne D, Dorigo M and Glover F 1999 *New Ideas in Optimization* (Mc-graw-Hill Publications, London)
8. Price K V and Storn R 1997 *Dr. Dobb’s J.* 22 18
9. Price K. and Storn R. 2014.. URL: http://www.ICS.Berkeley.edu/~storn/code.html.
10. Babu B V and Angira R 2006 *Comput. Che. Engng.* 30 989
11. Gujarathi AM and Babu BV 2009 *Chem. Eng. Sci.* 65 2009
12. Gujarathi AM and Babu BV 2009 *Ind. Eng. Chem. Res.* 48 11115
13. Gujarathi AM and Babu BV 2010 *Int. J Bio-insp. Compu.* 2 213
14. Gujarathi AM and Babu BV 2009 *Mater. Manuf. Processes.* 24 303
15. Gujarathi AM and Babu BV 2011) *Mater. Manuf. Processes.* 26 455
16. Gujarathi AM and Babu BV 2011 App. Comp. Intel. Soft. Comp. in press.
17. Gujarathi AM, Babu BV (2011) *Comput. Meth. Mater. Sci.* 11 463
18. Ripon KSN, Kwong S and Man KF 2007 *Info. Sci.* 177 632
19. Kasat RB and Gupta SK 2003 *Comput. Che. Engng.* 27 1785
20. Sankarrao and Gupta SK 2007 *Comput. Che. Engng.* 31 1496
21. Agrawal N, Rangaiah GP, Ray AK and Gupta SK 2006 *Ind. Eng. Chem. Res.* 45 3182
22. Gujarathi AM and Purohit S 2014. Under preparation.
23. Gujarathi AM, Porohit S, Manachanda R and Ghosh D 2011 Differential Evolution using Jumping Genes Adaptation. *Proc. of Indian Int. Conf. on Artificial Intel.* (IICAI 2011), Bangalore, India.
24. Ryoo HS and Sahinidis NV 1995 *Comput. Che. Engng.* 19: 551–566.