Optimization Of Water Pumping System Using Differential Evolution Strategies

B.V. Babu* and Rakesh Angira
Department of Chemical Engineering,
Birla Institute of Technology & Science,
Pilani-333 031 (Rajasthan) India
(bvbabu, angira) @bits-pilani.ac.in

Abstract

Differential Evolution (DE) is a population based search algorithm that comes under the category of evolutionary optimization techniques. It is an improved version of GA, and is exceptionally simple, significantly faster & robust at numerical optimization and is more likely to find a function's true global optimum. In the present study, DE is used to solve the classical optimization problem of water pumping system. Comparison is made with Branch & Reduce algorithm. The results indicate that performance of DE is better than the Branch & Reduce algorithm.

1. Introduction

The optimization of non-linear constrained problems is relevant to chemical engineering practice [1, 2]. In recent years, evolutionary algorithms (EAs) have been applied to the solution of NLP in many engineering applications. The best-known algorithms in this class include Genetic Algorithms (GA), Evolutionary Programming (EP), Evolution Strategies (ES) and Genetic Programming (GP). There are many hybrid systems, which incorporate various features of the above paradigms and consequently are hard to classify, which can be referred just as EC methods [3]. They differ from the conventional algorithms since, in general, only the information regarding the objective function is required. EC methods have been applied to a broad range of activities in process system engineering including modeling, optimization and control. Differential Evolution (DE), developed by Price & Storn [4], is one of the best EC methods. This method provides one of the best genetic algorithms for solving the real-valued test function. The convergence speed of DE is very high.

In the present study, DE – an evolutionary computation method, has been used to solve the Water Pumping System problem. This problem arises from the area of chemical engineering, and represent difficult non-linear optimization problem, with equality constraints. Comparison is made with Branch & Reduce algorithm [5]. It is found that DE, an exceptionally simple evolutionary computation method, is significantly faster and yields the global optimum for a wide range of the key parameters.

2. Differential evolution

Differential Evolution [4] is an improved version of GA [6] for faster optimization. Unlike simple GA that uses binary coding for representing problem parameters, DE uses real coding of floating point numbers. Among the DE's advantages are its simple structure, ease of use, speed and robustness. Price & Storn [4] gave the working principle of DE with single strategy. Later on, they suggested ten different strategies of DE [7]. A strategy that works out to be the best for a given problem may not work well when applied for a different problem. Also, the strategy and key parameters to be adopted for a problem are to be determined by trial & error. The key parameters of control are: NP - the population size, CR - the crossover constant, F - the weight applied to random differential (scaling factor). The pseudo code for DE, used in the present study is as follows:

- *Input* the value of D, NP, CR, F, strategy number ("strategy"$\in \{1, 2, 3, \ldots\ldots, 10\}$) & gen_max and lower & upper bounds of variables ($x_j^{(lo)}, x_j^{(hi)}$).
- *Initialize* all the vector population randomly in the given upper & lower bound.
  count = 0; (*count* is generation counter)
  for i $\le$ NP and for $j \le D$:
  $$x_{i,j,count=0} = x_j^{lo} + rand_j[0,1] * \left(x_j^{(hi)} - x_j^{(lo)}\right)$$
  End for.
- *Checking of constraints* (applying penalty if violated) and Evaluation of objective function.
  if first constraint violate
  pen= (constraint voilation) $* 10^4$;
  [Make violation +ve if it is –ve]
  $$f\left(x_{i,count}\right) = f\left(x_{i,count}\right) + pen;$$

Continue;
End *if.*
*if s*econd constraint violate
$pen = (\text{constraint voilation}) * 10^4$;
[Make violation +ve if it is –ve]
$f(x_{i,count}) = f(x_{i,count}) + pen$;
Continue;
End *if.*
.
.
if all constraints satisfy; $f(x_{i,count}) = f(x_{i,count})$;
End *if.*
End ***for.***

- *Find out* the vector with the *lowest cost*.
  Assign bestit$_j$ = best$_j$ = $x_{j,\ count}$ and $f_{\min}$ = $f(x_{i,count})$.
- *Repeat:*
  1. Perform *mutation & crossover*.
     For each target vector ($x_{j,count}$), select three distinct vectors (select five, if two vector differences are to be used) randomly from the current population (primary array) other than target vector. Randomly select *r*1, *r*2, *r*3, $\in \{1,2,3\ldots\ldots, \text{NP}\}$,
     except: $r1 \neq r2 \neq r3 \neq i$
     $j \in \{1,2,\ldots., D\}$ randomly selected for each i.

  if strategy = 1 i.e. DE/rand/1/bin
  for k ≤*D*:
  $u_{j,count+1} = x_{r3,j,count} + \text{F} * (x_{r1,j,count} - x_{r2,j,count})$
  if (rand$_j$ [0, 1]< CR or k = *D*)
  $u_{j,count+1} = x_{i,j,count}$ Otherwise
  If bounds are violated:
  $u_{j,count+1} = x_j^{lo} + rand_j[0,1] * (x_j^{(hi)} - x_j^{(lo)})$
  $if\ (u_{j,count+1} < x_j^{lo}$ or $u_{j,count+1} > x_j^{(hi)})$
  $u_{j,count+1} = u_{j,count+1}$ Otherwise
  End for.
  End ***if.***
  .
  .
  . (Till 10th strategy)

  2. *Check constraint violation* (apply penalty if violated) and perform *selection* for each target vector by comparing its cost with that of the trial vector. Vector with lower cost is selected for next generation.
  if first constraint violate
  $pen = (\text{constraint voilation}) * 10^4$;
  [make violation +ve if it is –ve]
  $f(u_{j,count+1}) = f(u_{j,count+1}) + pen$;
  Select;
  best$_j$ = $u_{j,count+1}$ if $f(u_{j,count+1}) \leq f_{\min}$
  Continue;
  End *if.*

  if second constraint violate
  $pen = (\text{constraint voilation}) * 10^4$;
  [make violation +ve if it is –ve]
  $f(u_{j,count+1}) = f(u_{j,count+1}) + pen$;
  Select;
  best$_j$ = $u_{j,count+1}$ if $f(u_{j,count+1}) \leq f_{\min}$

  Continue;
  End *if.*
  .
  .
  .
  if all constraints satisfy
  $f(u_{j,count+1}) = f(u_{j,count+1})$;
  Select;
  best$_j$ = $u_{j,count+1}$ if $f(u_{j,count+1}) \leq f_{\min}$

  Continue;
  End *if.*
  Where **Select** is:
  $$x_{i,count+1} = \begin{cases} u_{i,count+1} & \text{if } f(u_{i,count+1}) \leq f(x_{i,count}) \\ x_{i,count} & \text{otherwise} \end{cases}$$
  End ***for***.
  bestit$_j$ = best$_j$.
  Find maximum and minimum value of '*f*'.
  count=count+1;
  Terminate if $f_{\max} - f_{\min} \leq \varepsilon$ (10^{-6} or 10^{-7}).
  (As per desired accuracy)
- Till *termination criteria* do not meet.
- Print results.

The crucial idea behind DE is a scheme for generating trial parameter vectors. Basically, DE adds the weighted difference between two population vectors to a third vector. Price & Storn [7] have given some simple rules for choosing key parameters of DE for any given application. DE has been successfully applied in various fields. The various applications of DE are: digital filter design [8], batch fermentation process [9, 10], estimation of heat transfer parameters in trickle bed reactor [11], optimal design of heat exchangers [12,13], synthesis & optimization of heat integrated distillation system [14], optimization of an alkylation reaction [15], scenario-integrated optimization of dynamic systems [16], optimization of non-linear functions [17], optimization of thermal cracker operation [18], a differential evolution approach for global optimization of MINLP problems [19], Optimization of Non-Linear Chemical Processes [20], Evolutionary Computation for Global Optimization of Non-Linear Chemical Engineering Processes [21] etc.

3. The Problem

A water pumping system [22] consists of two parallel pumps drawing water from a lower reservoir and delivering it to another that is 40 m higher, as shown in fig 1. In addition to overcoming the pressure difference due to the elevation, the friction in the pipe is $7.2w^2$ kPa, where w is the combined flow rate in kilograms per second. The pressure-flow-rate characteristics of the pumps are:

Pump 1: $\Delta p\ (\text{kPa}) = 810 - 25w_1 - 3.75w_1^2$

Pump 2: $\Delta p\ (\text{kPa}) = 900 - 65w_2 - 30w_2^2$

where w_1 and w_2 are the flow rates through pump 1 and pump 2, respectively.

The system can be represented by four simultaneous equations. The pressure difference due to elevation and friction is:

$$\Delta p = 7.2w^2 + \frac{(40\ \text{m})(1000\ \text{kg/m}^3)(9.807\ \text{m/s}^2)}{1000\ \text{Pa/kPa}} \qquad (1)$$

$$\text{Pump 1: } \Delta p = 810 - 25w_1 - 3.75w_1^2 \qquad (2)$$

$$\text{Pump 2: } \Delta p = 900 - 65w_2 - 30w_2^2 \qquad (3)$$

$$\text{Mass balance: } \quad w = w_1 + w_2 \qquad (4)$$

The objective here is to minimize Δp subject to the constraints (1), (2), (3), and (4). Hence,

$$\textit{Min.}\ \Delta p = 7.2w^2 + \frac{(40\ \text{m})(1000\ \text{kg/m}^3)(9.807\ \text{m/s}^2)}{1000\ \text{Pa/kPa}}$$

Subject to:

$$\Delta p = 810 - 25w_1 - 3.75w_1^2 .$$

$$\Delta p = 900 - 65w_2 - 30w_2^2 .$$

$$w = w_1 + w_2 .$$

Stoecker [22] used the method of successive substitution for solving this problem.

4. Problem Modification

Liebman et al. [23] modified the above problem as given below:

$$\textit{Min.} \quad f = x_3 ,$$

Subject to:

$$x_3 = 250 + 30x_1 - 6x_1^2$$

$$x_3 = 300 + 20x_2 - 12x_2^2$$

$$x_3 = 150 + 0.5(x_1 + x_2)^2$$

$$0 \le \mathbf{x} \le (9.422, 5.903, 267.42)$$

Ryoo & Sahinidis [5] solved this problem using Branch and Reduce algorithm. They used different strategies of Branch and Reduce algorithm. The CPU-time reported by them ranges from a minimum of 0.3 s to a maximum of 150 s for various strategies used by them on Sun SPARC station 2. However author [5] did not mention about configuration of the system used but from site http://www.jinr.dubna.su/unixinfo/sun/suns1.html, the available configuration for Sun SPARC station 2 is: CPU - (1x 40MHz), RAM - 32MB and Hard Disk - 1.2Gb. The termination criterion used was an accuracy (ε) =10^{-6}. The global optimum reported is (**x**; *f*) = (6.293429, 3.821839, 201.159334; 201.159334).

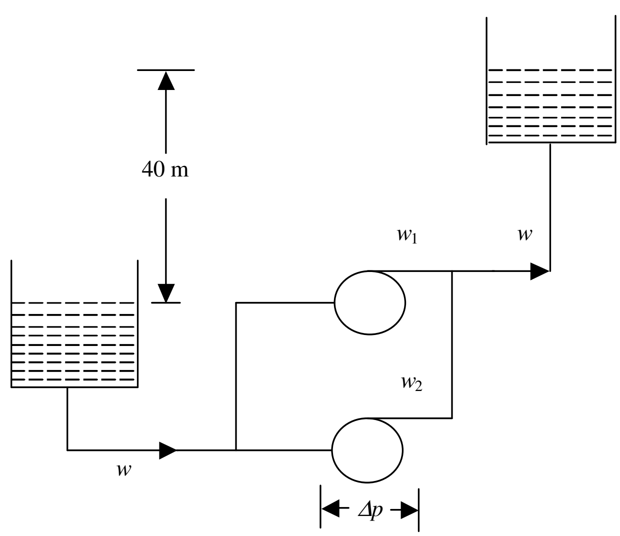

Fig 1. Water Pumping System

5. Problem Reformulation

In general, the equality constraints are difficult to deal with. And, more so in the case of evolutionary algorithms. So there is a need to transform equality constraints into inequality constraints by some means or the other. Typically, they are handled by either of the following two methods, viz., (1) eliminating the parameter and hence reducing the dimensions of the problem (2) an equality constraint is formulated into two inequalities by introducing deviation variables on problem parameter. In the present study, one variable is eliminated while the other two equalities are transformed into inequalities using method 1. Hence, the reformulated problem is as follows:

$$Min.\ \boldsymbol{f} = x_3 = 150 + 0.5(x_1 + x_2)^2$$

Subject to:

$$6x_1^2 - 30x_1 - 249.9999999 + 150.0 + 0.5(x_1 + x_2)^2 \geq 0.0$$

$$12x_2^2 - 20x_2 - 299.9999999 + 150.0 + 0.5(x_1 + x_2)^2 \geq 0.0$$
$$0 \leq \mathbf{x} \leq (9.422, 5.903)$$

The global optimum obtained is: (**x;** ***f***) **=** (6.293429, 3.821839; 201.159334).

6. Results & Discussion

Table 1 shows the results obtained by both DE and Branch & Reduce algorithm [5]. It may be noted that the global optimum is same as reported in [5] i.e. f = 201.159334 and the flow rates through Pump 1 & Pump 2 are x_1 = 6.293430 & x_2 = 3.821839 respectively.

Table 1. Comparison of DE with Branch & Reduce

Parameters	**Using DE**	**Using Branch & Reduce**
(x_1)	6.293430	6.293429
(x_2)	3.821839	3.821839
$(x_3) = f$ (i.e. same as the objective function)	201.159334	201.159334
CPU-time (s)	0.0714*	0.3$
Objective function (f)	201.159334	201.159334

* CPU-time on PC with Pentium PIII, 500 MHz/128 MB RAM/ 10 Gb HD with strategy no. 10

$ CPU-time on Sun SPARC Station 2

Table 2 & 3 presents the comparison, in terms of the number of objective function evaluations, CPU-time and proportion of convergences to the optimum, between the different DE strategies. The termination criterion used is accuracy of 10^{-6} and 10^{-7} respectively. In these tables, *NFE, NRC* and CPU-time represents, respectively the mean number of objective function evaluations over all the 10 executions, the percentage of convergences to the global optimum and the average CPU time per execution (key parameters used are NP = 20, CR = 0.5, F = 0.8).

Table 2. Results of DE with all ten strategies (accuracy (ε)= 10^{-6}).

S. No.	Strategy	*NFE*	CPU- time (s)	*NRC*
1	DE/rand/1/bin	3134	0.1319	100
2	DE/best/1/bin	2406	0.0879	100
3	DE/best/2/bin	4444	0.1758	100
4	DE/rand/2/bin	4644	0.1758	100
5	DE/rand-to-best/1/bin	2364	0.0879	100
6	DE/rand/1/exp	3214	0.1154	100
7	DE/best/1/exp	2372	0.0934	100
8	DE/best/2/exp	4506	0.1648	100
9	DE/rand/2/exp	4652	0.1868	100
10	DE/rand-to-best/1/exp	2162	0.0714	100

Table 3. Results of DE with all ten strategies [accuracy (ε)= 10^{-7}].

S. No.	Strategy	*NFE*	CPU- time (s)	*NRC*
1	DE/rand/1/bin	3524	0.1374	100
2	DE/best/1/bin	2624	0.1099	100
3	DE/best/2/bin	5016	0.1868	100
4	DE/rand/2/bin	5158	0.2033	100
5	DE/rand-to-best/1/bin	4146	0.1648	90
6	DE/rand/1/exp	3542	0.1319	100
7	DE/best/1/exp	2636	0.0989	100
8	DE/best/2/exp	5048	0.1923	100
9	DE/rand/2/exp	5206	0.1978	100
10	DE/rand-to-best/1/exp	2484	0.0989	100

The time taken by DE is much less than that of Branch & Reduce algorithm (Table-1). Of course the CPU-times cannot be compared directly because different computers are used. From the above table-2 & 3 it is evident that the strategy number 10 (DE/rand-to-best/1/exp) is the best strategy. It takes least average CPU-time, maximum *NRC* and minimum *NFE*.

7. Conclusion

In this paper the optimization of water pumping system using Differential evolution (DE) has been presented. The key parameters used for the present problem are: NP = 40, CR = 0.8, F = 0.5. The strategy that took minimum CPU-time with highest NRC is strategy no. 10 (DE/rand-to-best/1/exp). The results obtained by two methods (viz. DE & Branch & Reduce algorithm) are same and matches with that reported in literature.

Nomenclature

NFE	mean number of objective function evaluations
NRC	the percentage of convergencies to the global optimum
F	Scaling Factor
CR	The cross-over constant
D	dimension of the problem
gen_max	Maximum number of generation (1000 in the present study)
pen	Penalty in case of constraint violation
NP	Population size
$bestit_j$	best vector of the generation
$best_j$	current best vector
$u_{j,count+1}$	Child vector/ trial vector
$x_{i,count+1}$	Vector for next generation

References

[1] R. L. Salcedo, Solving Nonconvex Nonlinear Programming Problems with Adaptive Random Search. *Industrial & Engineering Chemistry Research,* **31**, 262, 1992.

[2] C. A. Floudas, *Nonlinear and mixed-integer optimization*. Oxford University Press, New York, 1995.

[3] D. Dasgupta, and Z. Michalewicz, *Evolutionary algorithms in Engineering Applications,* 3 - 23, Springer, Germany, 1997.

[4] K. Price, and R. Storn, Differential Evolution - A simple evolution strategy for fast optimization. *Dr. Dobb's Journal,* **22** (4)*,* 18 – 24 and 78, 1997.

[5] H. S. Ryoo, and N. V. Sahinidis, Global optimization of nonconvex NLPs and MINLPs with Applications in Process Design. *Computers & Chemical Engineering,* **19**(5), 551-566, 1995.

[6] D. E. Goldberg, *Genetic Algorithms in search, Optimization, and Machine learning,* Reading, MA, Addison-Wesley, 1989.

[7] K. Price, and R. Storn, *Web site of DE as on July, 2003,* the URL of which is: http://www.ICSI.Berkeley.edu/~storn/code.html

[8] R. Storn, Differential Evolution design of an IIR-filter with requirements for magnitude and group delay. *International Computer Science Institute,* TR-95-026, 1995.

[9] J. P. Chiou, and Wang, F. S., Hybrid method of evolutionary algorithms for static and dynamic optimization problems with application to a fed-batch fermentation process. *Computers & Chemical Engineering,* **23**, 1277-1291, 1999.

[10] F. S. Wang, and W. M. Cheng, Simultaneous optimization of feeding rate and operation parameters for fed-batch fermentation processes. *Biotechnology Progress,* **15** (5), 949-952, 1999.

[11] B. V. Babu, and K. K. N. Sastry, Estimation of heat-transfer parameters in a trickle-bed reactor using differential evolution and orthogonal collocation. *Computers & Chemical Engineering,* **23**, 327–339, 1999.

[12] B. V. Babu, and S. A. Munawar, Differential Evolution for the optimal design of heat exchangers. *Proceedings of All-India seminar on Chemical Engineering Progress on Resource Development: A Vision 2010 and Beyond*, IE (I), Bhubaneswar, India, March 11, (2000).

[13] B. V. Babu, and S. A. Munawar, Optimal Design of Shell & Tube Heat Exchanger by Different strategies of Differential Evolution. *PreJournal.com - The Faculty Lounge, Article No. 003873, posted on website Journal http://www.prejournal.com.* 2001.

[14] B. V. Babu, and R. P. Singh, Synthesis & optimization of Heat Integrated Distillation Systems Using Differential Evolution. *Proceedings of All-India seminar on Chemical Engineering Progress on Resource Development: A Vision 2010 and Beyond*, IE (I), Bhubaneswar, India, March 11, 2000.

[15] B. V. Babu, and C. Gaurav, Evolutionary Computation strategy for Optimization of an Alkylation Reaction. *Proceedings of International Symposium & 53rd Annual Session of IIChE* (*CHEMCON-2000*), Calcutta, December 18-21, 2000.

[16] B. V. Babu, and K. Gautam, Evolutionary Computation for Scenario-Integrated optimization of Dynamic Systems. *Proceedings of International Symposium & 54th Annual Session of IIChE* (*CHEMCON-2001*), Chennai, December 19-22, 2001.

[17] B. V. Babu, and R. Angira, Optimization of Non-linear functions using Evolutionary Computation. *Proceedings of 12th ISME Conference*, India, January 10–12, 153-157, 2001.

[18] B. V. Babu, and R. Angira, Optimization of thermal cracker operation using Differential Evolution. *Proceedings of International Symposium & 54th Annual Session of IIChE* (*CHEMCON-2001*), Chennai, December 19-22, 2001.

[19] B. V. Babu, and R. Angira, 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, pp. 880-884, 2003.

[20] B. V. Babu, and R. Angira, Optimization of Non-Linear Chemical Processes Using Evolutionary Algorithm. Proceedings of International Symposium & 55th Annual Session of IIChE (CHEMCON-2002), OU, Hyderabad, December 19-22, 2002.

[21] Rakesh Angira and B. V. Babu, Evolutionary Computation for Global Optimization of Non-Linear Chemical Engineering Processes". Proceedings of International Symposium on Process Systems Engineering and Control (ISPSEC' 03) - For Productivity Enhancement through Design and Optimization, IITBombay, Mumbai, January 3-4, 2003, Paper No. FMA2, pp 87-91, 2003.

[22] W. F. Stoecker, *Design of Thermal Systems*. 3rd ed., McGraw-Hill International edition, Singapore, pp 117-121, 1971

[23] J. Leibman, L. Lasdon, L. Schrage and A. Waren, Modeling and optimization with GINO. *The Scientific Press*, Palo Alto, CA, 1986.

Note: References [11, 12, 13, 15, 16, 17, 18, 19, 20, and 21] are also available via Internet as .pdf file at *http://bvbabu.50megs.com/custom.html/*[#24, 28, 35, 31, 39, 34, 38, 46, 54, 56 respectively].

And, references [11, 13, 15, 16, 18] are also available at Homepage of DE, the URL of which is: http://www.icsi.berkeley.edu/~storn/code.html as Application No. 13, 18, 19, 21, and 20 respectively.