MODELING AND OPTIMIZING OXYGEN PRODUCTION SYSTEM USING DIFFERENTIAL EVOLUTION

M. F. Tabassum[1]*, M. Saeed[1], N. A. Chaudhry[2], J. Ali [1], A. Sana[1]

[1]Department of Mathematics, University of Management and Technology, Lahore, Pakistan.
[2]Department of Mathematics, UCEST, Lahore Leads University, Lahore, Pakistan.
* Corresponding Author: Muhammad Farhan Tabassum, farhanuet12@gmail.com, +92-321-4280420

ABSTRACT: *In this article a non-linear optimization model of Oxygen production system is formulated. The production rate, pressure in storage tank, compressor power and storage tank volume are considered as the constraints of the model. The constraints in the formulated Oxygen production system model are handled by using 2-parameter-exponential penalty function. A novel optimization method based on a recently introduced Evolutionary Algorithm called Differential Evolution is described. Oxygen production system model is selected to demonstrate the capabilities and practical use of the method. The novel method is found easy to implement effectively, efficient and robust. The results obtained by this technique make it an attractive applicable approach for solving design problems in various engineering disciplines. The results are compared with previous studies also.*

Key Words: Optimization, Differential evolution, 2-parameter-exponential penalty function, Oxygen production system.

(Presented at the 5th International. Multidisciplinary Conference, 29-31 Oct., at, ICBS, Lahore)

1. INTRODUCTION

In simple terms, optimization is the attempt to maximize a system's desirable properties while simultaneously minimizing its undesirable characteristics. What these properties are and how effectively they can be improved depends on the problem at hand [1]. Price and Storn developed DE to be a reliable and versatile function optimizer that is also easy to use [2].

DE is a population-based optimizer that attacks the starting point problem by sampling the objective function at multiple, randomly chosen initial points. Preset parameter bounds define the domain from which the *Np* vectors in this initial population [3]. Each vector is indexed with a number from 0 to *Np-1*. Like other population-based methods, DE generates new points that are perturbations of existing points, but these deviations are neither reflections like those in the CRS [4] and Nelder–Mead methods [5], nor samples from a predefined probability density function, like those in the ES [6]. Instead, DE perturbs vectors with the scaled difference of two randomly selected population vectors. To produce the trial vector, $\mathbf{u}_0$, DE adds the scaled, random vector difference to a third randomly selected population vector [7]. In the selection stage, the trial vector competes against the population vector of the same index, which in this case is number. The select-and-save step in which the vector with the lower objective function value is marked as a member of the next generation. Indicate that the procedure repeats until all *Np* population vectors have competed against a randomly generated trial vector. Once the last trial vector has been tested, the survivors of the *Np* pairwise competitions become parents for the next generation in the evolutionary cycle [8].

To change the constrained optimization into un-constrained one, by adding or subtracting the values from the objective functions is reported by [9-12].

In this script a mathematical model of Oxygen production system for minimum cost is reformulated and selected as a test case for the capabilities of DE methods. So far no such applications of these methods to such a challenging engineering optimization problem have been found. For selecting the best method there is a necessity to conduct comparative studies of their potential applications to modern world problems, like the one formulated in this study.

2. MATERIALS AND METHODS

The motivation for this research was to modify Oxygen production model. The Evolutionary method was used for the optimization of Oxygen production model. This method was basically designed for unconstrained optimization problems. In formulated optimization Oxygen production model the constraints were handled by 2-parameter-exponential penalty function.

2.1. Differential Evolution

Differential Evolution's (DE) most versatile implementation maintains a pair of vector populations, both of which contain *Np* *D*-dimensional vectors of real-valued parameters. Differential evolution consist of the following steps [13, 14]:

Initialization: The current population, symbolized by $\boldsymbol{P_x}$, is composed of those vectors, $\boldsymbol{x_{i,g}}$, that have already been found to be acceptable either as initial points, or by comparison with other vectors:

$$P_{x,g} = (x_{i,g}),\ i = 0,1,2,\ldots, Np-1,\ g = 0,1,2,\ldots, g_{\max} \quad (1)$$

$$x_{i,g} = (x_{j,i,g}), \quad j = 0,1,2,\ldots, D-1,$$

Indices start with ***0*** to simplify working with arrays and modular arithmetic. The index, $\boldsymbol{g = 0, 1,\ldots, g_{max}}$, indicates the generation to which a vector belongs. In addition, each vector is assigned a population index, ***i***, which runs from ***0*** to ***Np −* 1**. Parameters within vectors are indexed with ***j***, which runs from ***0*** to ***D −* 1**.

Before the population can be initialized, both upper and lower bounds for each parameter must be specified. These ***2D*** values can be collected into two, ***D***-dimensional initialization vectors, $\boldsymbol{b_L}$ and $\boldsymbol{b_U}$, for which subscripts ***L*** and ***U*** indicate the lower and upper bounds, respectively. Once initialization bounds have been specified, a random number generator assigns each parameter of every vector a value from within the prescribed range. For example, the initial value ***(g = 0)*** of the ***j***th parameter of the ***i***th vector is

$$x_{j,i,0} = rand_j(0,1).(b_{j,U} - b_{j,L}) + b_{j,L}. \quad (2)$$

The random number generator, $\boldsymbol{rand_j}$ ***(0, 1)***, returns a uniformly distributed random number from within the range ***[0, 1)***, i.e., $\boldsymbol{0 \le rand_j\ (0, 1) < 1}$. The subscript, ***j***, indicates that a new random value is generated for each parameter. Even if a variable is discrete or integral, it should be initialized with a real value since DE internally treats all variables as floating-point values regardless of their type.

Mutation: Once initialized, DE mutates randomly chosen vectors to produce an intermediary population, $\boldsymbol{P_{v,g}}$, of ***Np*** mutant vectors, $\boldsymbol{v_{i,g}}$:

$$P_{v,g} = \left(v_{i,g}\right),\ i = 0,1,2,...,Np-1,\ g = 0,1,2,,...,g_{\max} \quad (3)$$
$$v_{i,g} = \left(v_{j,i,g}\right), \quad j = 0,1,2,...,D-1,$$

DE mutates and recombines the population to produce a population of ***Np*** trial vectors. In particular, differential mutation adds a scaled, randomly sampled, vector difference to a third vector. Eq. 4 shows how to combine three different, randomly chosen vectors to create a mutant vector, $\boldsymbol{v_{i,g}}$:

$$v_{i,g} = x_{r0,g} + F.\left(x_{r1,g} - x_{r2,g}\right). \quad (4)$$

The scale factor, ***F ϵ (0, 1+)***, is a positive real number that controls the rate at which the population evolves. While there is no upper limit on ***F***, effective values are seldom greater than ***1.0***. The base vector index, $\mathbf{r_0}$, can be determined in a variety of ways, but for now it is assumed to be a randomly chosen vector index that is different from the target vector index, ***i***. Except for being distinct from each other and from both the base and target vector indices, the difference vector indices, $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$, are also randomly selected once per mutant. Fig.1 illustrates how to construct the mutant, $\boldsymbol{v_{i,g}}$, in a two-dimensional parameter space.

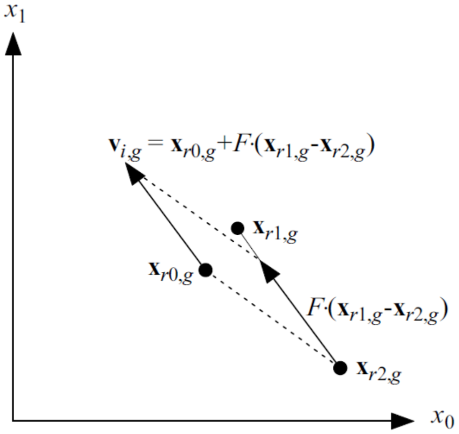

x_1
$\mathbf{v}_{i,g} = \mathbf{x}_{r0,g} + F\cdot(\mathbf{x}_{r1,g}\text{-}\mathbf{x}_{r2,g})$
$\mathbf{x}_{r1,g}$
$\mathbf{x}_{r0,g}$
$F\cdot(\mathbf{x}_{r1,g}\text{-}\mathbf{x}_{r2,g})$
$\mathbf{x}_{r2,g}$
x_0

Fig-1: Differential mutation: the weighted differential, *F* (*$x_{r1,g}$*– *$xr_{2,g}$*), is added to the base vector, *$x_{r0,g}$*, to produce a mutant, *$v_{i,g}$*.

Crossover: Each vector in the current population is then recombined with a mutant to produce a trial population, ***Pu***, of ***Np*** trial vectors, $\boldsymbol{u_{i,g}}$:

$$P_{u,g} = \left(u_{i,g}\right),\ i = 0,1,2,...,Np-1,\ g = 0,1,2,,...,g_{\max} \quad (5)$$
$$u_{i,g} = \left(u_{j,i,g}\right), \quad j = 0,1,2,...,D-1,$$

During recombination, trial vectors overwrite the mutant population, so a single array can hold both populations. To complement the differential mutation search strategy, DE also employs uniform crossover. Sometimes referred to as discrete recombination, (dual) crossover builds trial vectors out of parameter values that have been copied from two different vectors. In particular, DE crosses each vector with a mutant vector:

$$u_{i,g} = u_{j,i,g} = \begin{cases} v_{j,i,g} & if\left(rand_j(0,1) \le Cr \ or\ j = j_{rand}\right) \\ x_{j,i,g} & otherwise \end{cases} \quad (6)$$

The crossover probability, ***Cr ϵ [0, 1]***, is a user-defined value that controls the fraction of parameter values that are copied from the mutant. To determine which source contributes a given parameter, uniform crossover compares ***Cr*** to the output of a uniform random number generator, ***$rand_j$ (0, 1)***. If the random number is less than or equal to ***Cr***, the trial parameter is inherited from the mutant, $\boldsymbol{v_{i,g}}$; otherwise, the parameter is copied from the vector, $\boldsymbol{x_{i,g}}$. In addition, the trial parameter with randomly chosen index, $\boldsymbol{j_{rand}}$, is taken from the mutant to ensure that the trial vector does not duplicate $\boldsymbol{x_{i,g}}$. Because of this additional demand, ***Cr*** only approximates the true probability, $\boldsymbol{p_{Cr}}$, that a trial parameter will be inherited from the mutant. Fig. 2 plots the possible trial vectors that can result from uniformly crossing a mutant vector, $\boldsymbol{v_{i,q}}$, with the vector $\boldsymbol{x_{i,q}}$.

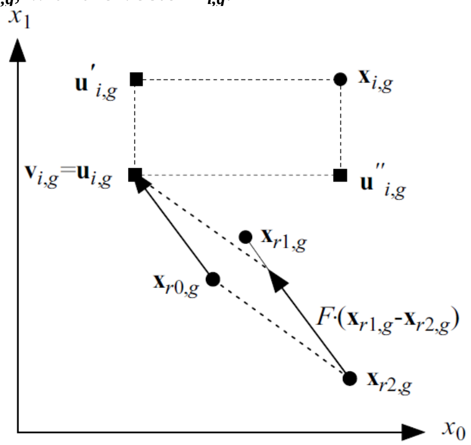


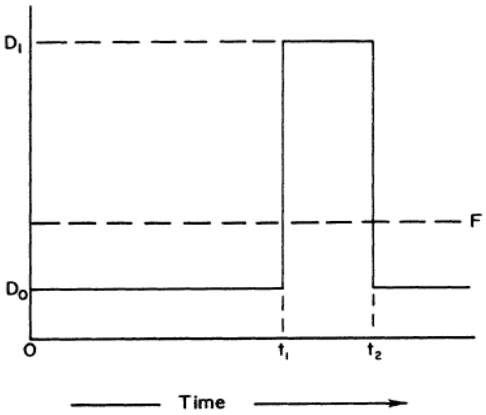
Fig-2: The possible additional trial vectors *$u'_{i,g}$* , *$u''_{i,g}$* when *$x_{i,g}$* and *$v_{i,g}$* are uniformly crossed.

Selection: If the trial vector, $\boldsymbol{u_{i,g}}$, has an equal or lower objective function value than that of its target vector, $\boldsymbol{x_{i,g}}$, it replaces the target vector in the next generation; otherwise, the target retains its place in the population for at least one more generation Eq. 7. By comparing each trial vector with the target vector from which it inherits parameters, DE more tightly integrates recombination and selection than do other EAs:

$$x_{i,g+1} = \begin{cases} u_{i,g} & if \quad f\left(u_{i,g}\right) \le \left(x_{i,g}\right) \\ x_{i,g} & otherwise \end{cases} \quad (7)$$

Once the new population is installed, the process of mutation, recombination and selection is repeated until the optimum is located, or a pre-specified termination criterion is satisfied, e.g., the number of generations reaches a preset maximum, $\boldsymbol{g_{max}}$.

2.2. Formulation of Oxygen production system: In this problem the prime objective was to minimize the cost of oxygen furnace. This oxygen furnace was used in chemical reactor for the supply of pure oxygen. Oxygen production system [15] contained oxygen plant, compressor and storage tank for oxygen furnace. Different kinds of variables were assigned and different kinds of constraints were generated, therefore the oxygen demanded varied with respect to time interval shown in Fig. 3. Here $\boldsymbol{t_1}$ was time interval for rate of low demand $\boldsymbol{D_0}$ and $\boldsymbol{t_2}$ **-** $\boldsymbol{t_1}$ time for rate of high demand $\boldsymbol{D_1}$. Oxygen plants were designed to provide oxygen at a fixed rate.



$$D = \begin{cases} D_o & for \quad O \le t \le t_1 \\ D_1 & for \quad t_1 \le t \le t_2 \end{cases}$$

Fig-3: Cycle of demand

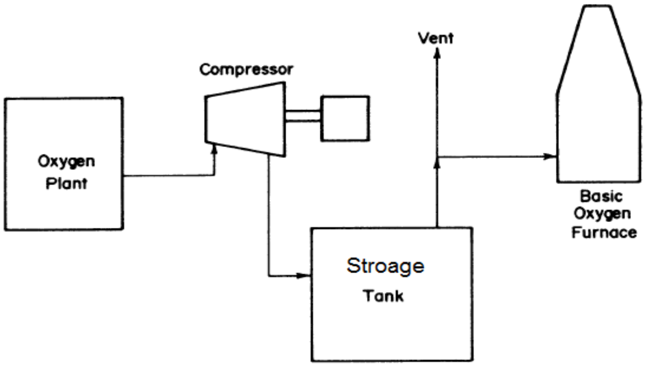


Fig-4: Oxygen production design

The capacity of oxygen plant = $\boldsymbol{D_1}$.

Assumptions: Oxygen furnace and demand cycle were fixed, no external factors were imposed, storage tank had standard design and compression of ideal gas was isothermal.

Total annual cost = oxygen production cost + compressor operating cost + compressor cost + storage vessel

The model consisted on design equations that narrated independent variables

Independent variables:

Oxygen plant production rate ***F***,
The compressor ***H***,
Storage tank design capacities ***V***,
The maximum tank pressure ***p***.

$\boldsymbol{I_{max}}$ = maximum stored oxygen

By using law of corrected gas as:

$$V = \frac{I_{max}}{M}\frac{RT}{p}z$$

where

***R** = gas constant,*

***T** = gas temperature,*

***ƶ** = compressibility factor,*

***M** = molecular weight of Oxygen.*

From Fig. 3, maximum oxygen = area under the demand curve between $\boldsymbol{t_1}$ and $\boldsymbol{t_2}$ and $\boldsymbol{D_1}$ and ***F***. Thus,

$$I_{max} = (D_1 - F)(t_2 - t_1) \tag{8}$$

Put the value I_{max} in above equation

$$V = \frac{(D_1 - F)(t_2 - t_1)}{M}\frac{RT}{p}z \tag{9}$$

As we now that the gas flow rate = $\frac{(D_1 - F)(t_2 - t_1)}{t_1}$

$$H = \frac{(D_1 - F)(t_2 - t_1)}{t_1}\frac{RT}{k_1 k_2} ln\left(\frac{p}{\rho_0}\right) \tag{10}$$

where

$\boldsymbol{k_1}$ = unit conversion factor,

$\boldsymbol{k_2}$ = compressor efficiency,

$\boldsymbol{\rho_0}$ = oxygen delivery pressure.

Rate of Oxygen plant *F* was sufficient, to supply the total demand of oxygen

$$F \ge \frac{(D_0 t_1) + D_1(t_2 - t_1)}{t_2} \tag{11}$$

Maximum pressure of tank > delivered Oxygen pressure

$$p \ge \rho_0 \tag{12}$$

Oxygen plant annual cost was

$$C_1\left(\frac{Rs}{year}\right) = a_1 + a_2 F \tag{13}$$

where $\boldsymbol{a_1}$ and $\boldsymbol{a_2}$ were empirical constants.

Empirical constants = *(fuel+ water+ labor)* costs for plants.

The capital cost for storage vessels

By using power correlation law, as

$$C_2(Rs) = b_1 V^{b_2} \tag{14}$$

where $\boldsymbol{b_1}$ and $\boldsymbol{b_2}$ were empirical constants.

Similarly capital compressors cost attained from a correlation was

$$C_3(Rs) = b_3 H^{b_4} \tag{15}$$

where $\boldsymbol{b_3}$ and $\boldsymbol{b_4}$ were empirical constants.

Whereas compressor power cost was approximately = $b_5 t_1 H$ where $\boldsymbol{b_5}$ was the power cost.

Total cost function

$$Annual\ cost = a_1 + a_2 F + d\,(b_1 V^{b_2} + b_3 H^{b_4}) + N b_5 t_1 H \tag{16}$$

where

N = number of cycles per year

d =annual cost factor.

To minimize eq. 16 represented complete design optimization problem that contained a suitable value of ***F, V, H***, and ***p***, cycle parameters were ***(N, D₀, D₁, t₁,** and **t₂)***, cost parameters were ***(a₁, a₂, b₁ to b₅,** and **d)*** and physical parameters were ***(T, ρ₀, k₂, ƶ** and **M)*** [16].

By using the new variables:

$\boldsymbol{z_1}$= production rate of oxygen plant,
$\boldsymbol{z_2}$= pressure in storage tank,
$\boldsymbol{z_3}$= compressor power and
$\boldsymbol{z_4}$= storage tank volume.

Non- linear programming model of oxygen design problem was as under

$$Minimizing\ Z = a_1 + a_2 z_1 + d(b_1 {z_3}^{b_2} + b_3 {z_4}^{b_4}) + N b_5 t_1 z_3$$

Subject to

$$t_2 z_1 \ge (D_0 t_1) + D_1(t_2 - t_1)$$

$$z_2 \ge \rho_0$$

$$z_3 = \frac{(D_1 - z_1)(t_2 - t_1)}{t_1}\frac{RT}{k_1 k_2} ln\left(\frac{z_2}{\rho_0}\right)$$

$$z_4 = \frac{RTz}{M}\frac{(D_1 - z_1)(t_2 - t_1)}{z_2}$$

$$z_1, z_2, z_3, z_4 \ge 0$$

Table-1: Different parameters for Oxygen supply system

Value of Cycle Parameters		
N	Number of cycles per year	1
D_o	Rate of low demand	2.5
D_1	Rate of High demand	40
t_1	Time for low demand	0.6
t_2	Time for High demand	2.5
Value of Cost Parameters		
a_1	Empirical constant (labor)	61.8
a_2	Empirical constant (fuel)	5.72
b_1	Empirical constant (vessel)	0.0175
b_2	Empirical constant (fitting)	0.85
b_3	Empirical constant (installed)	0.0094
b_4	Empirical constant (operating)	0.75
b_5	Power cost	0.006
d	Annual cost factor	1
Value of Physical Parameters		
T	Gas temperature	20^o
ρ_0	Oxygen delivery pressure	200
k_2	Compressor efficiency	0.5
M	Molecular weight of Oxygen	31.9999
z	Compressibility factor	28.2795

The final non-linear programming model of oxygen design problem became as

$$Minimizing\ Z = 61.8 + 5.72z_1 + (0.0175{z_3}^{0.85} + 0.0094{z_4}^{0.75}) + 0.0036z_3$$

Subject to

$$z_1 \geq 17.5$$

$$z_2 \geq 200$$

$$z_3 = 36.25\ \frac{(40-z_1)(0.4)}{0.6} ln\left(\frac{z_2}{200}\right)$$

$$z_4 = 348300 \frac{(40-z_1)(0.4)}{z_2}$$

$$z_1, z_2, z_3, z_4 \geq 0$$

2.3. Constrained Handling

The constrained problem is changed into unconstrained optimization problem using following penalty function [17]

$$G(x) = R_1\left\{e^{R_2\left(v^\delta\right)} - 1\right\}$$

where

$\boldsymbol{v}$ = max{0,g(x)}

$\boldsymbol{\delta}$ is such that 0< δ<1

$\boldsymbol{R_1}$, $\boldsymbol{R_2}$ are constants

$$v = \max\{0,\ z_1 - 17.5,\ z_2 - 200,\ z_3 - 36.25\ \frac{(40-z_1)(0.4)}{0.6} ln\left(\frac{z_2}{200}\right),\ z_4 - 348300\frac{(40-z_1)(0.4)}{z_2}\}$$

$$Minimizing\ Z = 61.8 + 5.72z_1 + (0.0175{z_3}^{0.85} + 0.0094{z_4}^{0.75}) + 0.0036z_3 + G(x).$$

3. RESULTS AND DISCUSSIONS

Evolutionary methods were popular because of their simplicity, flexibility, and reliability [18]. These methods have been shown to satisfy the first-order necessary conditions for a minimizer i.e., convergence to a stationary point [19]. In most of the Evolutionary methods a set of directions that span the search space was sufficient information to investigate the global and local behavior of the function [20].

As per study conducted by [16, 21] have reported the solution of the above formulated problem with different setting of parameters. The Oxygen production system was solved by using geometric programming approach considering smaller values of the parameters. The best solution of the problem also gave the minimum cost of $173.76 [21]. The same problem was solved by using gradient based method with a minimum cost of 173.83$ as reported by [16]. In this study the problem was solved by using DE method. The comparisons of the solutions found in this study are presented in table-2.

Table-2: Comparison of results

Sr. No.	Power cost $/(HP-HR)	Production Rate	Maximum Pressure	Minimum power cost			
				Jen [16]	**HJ**[12]	**NM**[12]	**DE**
1	0.0015	17.5	802.82062	172.21	**172.11700**	172.11705	**172.11700**
2	0.003	17.5	658.19221	172.85	172.74737	172.74745	**172.74734**
3	0.006	17.5	473.69271	173.83	173.74617	173.74621	**173.7415**
4	0.009	17.5	361.23119	174.52	174.45393	174.45394	**174.4539**
5	0.012	17.5	283.80233	175.95	174.91330	174.91335	**174.9133**
6	0.018	17.500001	200.00000	175.07	175.06747	175.06741	**175.0670**
7	0.024	17.500001	200	175.07	175.06	175.0601	**175.06**

The previous studies witnessed that NM method was comparatively a low computation cost method. On the other hand HJ method provided guaranteed convergence for a number of differentiable functions. It was observed that the solutions which are shown in table 2 DE method gives slightly better value than the other three results. These comparisons show that these methods like HJ, NM and DE were yet better choices for solving such exponential type optimization problems in engineering design but DE method was more reliable.

HJ method was terminated when the step length fell below 10^{-9}, NM was terminated when the maximum of 200×No of variables functions evaluation and DE was terminated when the maximum of 1000×No of variables functions evaluation were carried out. At these termination criteria the functions evaluation by DE method was very much smaller than all methods.

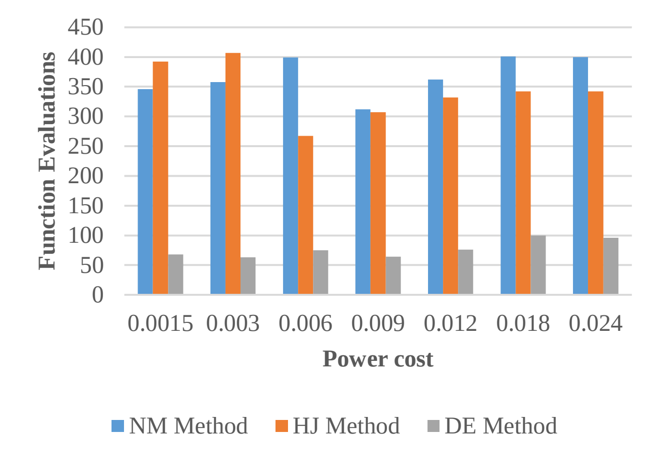


Fig-5: Comparison of function evaluations of NM, HJ and DE methods

It was concluded that on the radical objective functions like the modeled one, DE method was a better and low cast choice for power cost rate.

Table-3: Comparison of Function Evaluations between NM Method and HJ Method

Power cost	Function Evaluations		
	NM Method	HJ Method	DE Method
0.0015	346	392	**68**
0.003	358	407	**63**
0.006	399	267	**75**
0.009	312	307	**64**
0.012	362	332	**76**
0.018	401	342	**100**
0.024	400	342	**96**

The above table also shows that when the power cost was small, the number of functions evaluation of NM method was less than that of HJ method and when the power cost increased gradually the performance of HJ method was getting better than NM method, but the performance of DE method was exceptional well and better than both the methods.

For optimum results of Oxygen design problem, a general-purpose solver was required. For numerical simulation of the oxygen design model, the programming environment of MATLAB was found to be quite supportive due to availability of a plenty of built-in functions. Another important advantage of MATLAB was the fact that parameters were easily settled for handling constraints.

4. CONCLUSION

The outcome performances of Differential Evolution, Hooke-Jeeves and Nelder-Mead methods experimented via a number of initial guesses were carried out on formulated Oxygen production system. It was concluded that performance of DE method was better with respect to its efficiency of solving such a problem with minimum computational efforts as compared to those of HJ and NM methods. Through this work it is recommended that in any environment DE method is a better choice as compared to the class of methods involving HJ and NM methods.

REFERENCES

[1] Storn, R. and Price, K.V. “Differential Evolution – A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces”, *International Computer Science Insitute*, Berkeley, CA, (1995).

[2] Price, K.V., Storn, R.M. and Lampinen, J.A. “Differential Evolution: A Practical Approach to Global Optimization”, *Springer*, Berlin, (2005).

[3] Storn, R. “Differential Evolution Design of an IIR-Filter with Requirements for Magnitude and Group Delay”, *Technical Report TR-95-026*, International Computer Science Insitute, (1995).

[4] Storn, R. “Differential evolution design of an IIR-filter”, *1996 IEEE Int. Conf. Evolutionary Computation*, Nagoya, pp. 268–273, (1996).

[5] Nelder, J. A. and R. Mead. “A simplex method for function minimization”, *Comput. J.*, **7**:308–313 (1965).

[6] Johnson, I.R., Kinghorn, B.P., Murphy, S.R., Lodge, G.M. and Meszaros, S.A. “Estimating soil physical parameters using simulation differential evolution”, *IASTED Int. Conf. Applied Simulation Modelling*, Greece, pp. 274–279 (2002).

[7] Babu, B.V. and Angira, R. “Optimization of water pumping system using differential evolution strategies”, *2nd Int. Conf. Computational Intelligence Robotics Autonomous Systems*, Singapore, (2003).

[8] Nikolos, I.K. and Brintaki, A.N. “Coordinated UAV path planning using differential evolution”, *20th IEEE Int. Symp. Intelligent Control*, Limassol, Cyprus, (2005).

[9] Tabassum, M. F., M. Saeed , A. Sana, Nazir Ahmad, “Solution of 7 Bar Tress Model using Derivative Free Methods”, *Proceedings of the Pakistan Academy of Sciences*, **52**(3): 265-271 (2015).

[10] Tabassum, M.F., M. Saeed, Nazir Ahmad and A. Sana, “Solution of War Planning Problem Using Derivative Free Methods”, *Sci.Int.*, **27**(1): 395-398 (2015).

[11] Ali J., M. Saeed, N. A. Chaudhry and M. F. Tabassum. “New Mathematical Models of N-Queens Problem and its Solution by a Derivative-Free Method of Optimization”, *Sci.Int.*, **27**(3): 2005-2008 (2015).

[12] Tabassum, M. F., M. Saeed, N. A. Chaudhry, A. Sana and Z. Zafar., “Optimal Design of Oxygen Production System by Pattern Search Methods”, *Pakistan Journal of Science*, **67**(4): 371-376 (2015).

[13] Qing, A. “Dynamic differential evolution strategy and applications in electromagnetic inverse scattering problems”, *IEEE Transactions on Geoscience and Remote Sensing*, **44**(1): 116–125 (2006).

[14] Storn, R. “On the usage of differential evolution for function optimization”. *1996 Biennial Conf. North American Fuzzy Information Processing Society*, Berkeley, (1996).

[15] Ravindran, A., K. M. Ragsdell and G. V. Reklaitis. “Engineering Optimization Methods and Applications”. *John Wiley & Sons*; Canada, (2006).

[16] Jen, F. C., C. C. Pegels and T. M. Dupuis. “Optimal Capacities of Production Facilities”, *Manag. Sci.*, **14**:570–580 (1968).

[17] Chaudhry N. A., M. O. Ahmad and J. Ali. “Constraint Handling in Genetic Algorithms by A 2-Paramer-

Exponential Penalty Function Approach", *Pakistan Journal of Science*, **61**(3):122-129 (2009).

[18] Lewis, R.M., V. Torczon, M. W. Trosset, "Direct search methods: then and now", *Journal of Computational and Applied Mathematics*; **124**(1):191–207 (2000).

[19] Lucidi, S. and M. Sciandrone, "On the global convergence of derivative–free methods for unconstrained optimization", *SIAM J. Opt;* **13**: 97–116 (2002).

[20] Rios, L.M. and N. V. Sahinidis, "Derivative-free optimization: a review of algorithms and comparison of software implementations", *Journal of Global Optimization*, **56**(3):1247-1293 (2012).

[21] Hellinckx, L.J. and M.J. Rijckeart. "Optimal Capacities of Production Facilities: An Application of Geometric Programming", *The Canadian journal of Chemical Engineering*. **50**: 148-150 (1972).