A Novel Differential Evolution Algorithm with Q-Learning for Economical and Statistical Design of X-Bar Control Charts

Ahmad Abdulla Al-Buenain
Mechanical and Industrial Engineering Department
Qatar University
Doha, Qatar
laa1304017@student.qu.edu.qa

Damla Kizilay
Industrial Engineering Department
Izmir Democracy University
Izmir, Turkey
damla.kizilay@idu.edu.tr

Ozge Buyukdagli
The International University of Sarajevo, Department of Computer Science
Sarajevo, Bosnia Herzegovina
obuyukdagli@ius.edu.ba

M. Fatih Tasgetiren
Logistics Management Department,
Yasar University
Izmir, Turkey
fatih.tasgetiren@yasar.edu.tr

***Abstract*—This paper presents a novel differential evolution algorithm with Q-Learning (DE_QL) for the economical and statistical design of X-Bar control charts, which has been commonly used in industry to control manufacturing processes. In X-Bar charts, samples are taken from the production process at regular intervals for measurements of a quality characteristic and the sample means are plotted on this chart. When designing a control chart, three parameters should be selected, namely, the sample size (n), the sampling interval (h), and the width of control limits (k). On the other hand, when designing an economical and statistical design, these three control chart parameters should be selected in such a way that the total cost of controlling the process should be minimized by finding optimal values of these three parameters. In this paper, we develop a DE_QL algorithm for the global minimization of a loss cost function expressed as a function of three variables n, h, and k in an economic model of the X-bar chart. A problem instance that is commonly used in the literature has been solved and better results are found than the earlier published results.**

Keywords*—*Differential evolution, Q-learning, X-Bar control charts, Economical design of control charts.

I. Introduction

Statistical control charts are generally used to control manufacturing processes. The main objective of a control chart design is to detect the process shift by distinguishing between two different sources of variation in a process. These variations are called as assignable and common causes of variability [1]. In general, there are two types of control charts, i.e., charts for variables and charts for attributes. X-bar is a type of variable control chart, which is most widely employed in the industry because of its simplicity. The purpose of these charts is to determine the assignable causes leading to nonconforming products in manufacturing. When these assignable causes are determined, corrective actions can be taken before a large number of nonconforming products are manufactured. In addition, these methods also provide effective tools for determining the process parameters and making an analysis of process capability. As mentioned before, in the design of a control chart, three parameters should be determined. These are sample size n, sampling interval h, and width of control limits k for the chart. Selecting these three parameters is also known as the design of a control chart.

In general, control charts have been designed to minimize the two statistical errors, namely Type-I error (α) and Type-II error (β). However, in practice, the design of a control chart has some economical activities like sampling and testing, determining out-of-control signals, correcting and revising the out-of-control process, the loss of the company's goodwill on the delivering nonconforming products to customers and so on. For these reasons, the economical design of a control chart has been attracting more attention over recent years [2].

The economical design is a mathematical model where parameters of a control chart should be found by minimizing an expected cost function, which includes costs of sampling and testing, costs related to determining out-of-control signals and possibly correcting the assignable cause(s), and costs of allowing nonconforming units to customers. Duncan [3] first proposed an economic model for the design of the X-bar chart where they assumed that a random shift in the process means due to single assignable cause and the moving from in-control to the out-of-control state have an exponential distribution. Panagos et al. [4] defined two distinct situations in economic design (i) the process continues in operation while searches for the assignable cause are made and (ii) the process must be shut down during the search. Detailed literature reviews can be found in Montgomery [2], Svoboda [5] and Ho and Case [6] on the economic design of control charts where it is observed that the majority of the researchers have considered X-bar chart and Duncan's [3] single assignable cause model where the loss cost is expressed as a function of three variables n, h and k. Choosing these parameters on economic criteria is called economical design, and it is getting more and more popular due to its ability and capability of having the process under statistical control at lower cost [7–13]. The effectiveness of economic design depends on how accurately this loss cost function is minimized to determine the values of the three design variables. Several optimization techniques have been proposed to minimize these design variables [4–6,14]. However, very recently, some global optimization methods such as Genetic Algorithm [15], Particle Swarm Optimization [16,17], Simulated Annealing [18] have been developed to solve the problem on hand. tried for the same purpose. Some other global optimization algorithms have been also proposed for the economic design of charts other than the X-bar chart [19,12]. In addition to the above, teaching–learning-based optimization (TLBO) has been proposed for the minimization of the loss cost function in the economic design of the X-bar control chart. In this paper, we develop a DE_QL algorithm for the global minimization of a loss cost function expressed as a function of three variables n, h and k in an economic model of X-bar chart. A problem instance that is commonly used in the literature has been solved and slightly better results are found than the earlier published results.

The remainder of the paper is organized as follows: Section 2 briefly explains the economic model of the X-bar chart, i.e., the formulation of the loss cost function. The traditional DE algorithm is briefly given in Section 3. The DE_QL algorithm is described in Section IV. Finally, computational results are given in Section V.

II. FORMULATION OF COST FUNCTION

The loss cost function formulated by van Deventer and Manna [7], based on the economic model of Lorenzen and Vance [14] has been considered for optimization by DE_QL in the present work. It is also considered in [20]. A brief description of the formulation is given below. Initially, it is assumed that the process will be the in-control state. And, it is also assumed that the quality characteristic of the process to be normally distributed with mean 0 and variance σ^2. Suppose that the process is randomly disturbed because of the occurrence of an assignable cause at a rate of λ with an exponential distribution. With sample size and the width of control limits for the X-Bar chart are n and k, respectively, μ_0 will be the centerline and the two control limits will be defined as $\mu_0 \pm k\sigma/\sqrt{n}$. When the process is in-control, false alarm occurrences will be at a rate of

$$\alpha = 2\int_k^{\infty} \phi(z)dz \tag{1}$$

where $\phi(z)$ is the standard normal density.

Then, the shift in the process mean will be $\delta\sigma$, and the probability of the shift that will be detected on any sample will be as follows:

$$1-\beta = \int_{-\infty}^{(-k-\delta\sqrt{n})} \phi(z)dz + \int_{(-k-\delta\sqrt{n})}^{\infty} \phi(z)dz \tag{2}$$

While the process is in control, the expected number of samples, s, will be $s = \sum_{i=0}^{\infty} iP$ where iP is the assignable cause happens between the ith and (i+1)st sample. For this reason, s can be rewritten as

$$s = \sum_{i=0}^{\infty} i\left(e^{\lambda hi} - e^{\lambda h(i+1)}\right) = \frac{1}{e^{\lambda h}-1} \tag{3}$$

Besides, the average time of occurrence of the assignable cause within the ith and (i + 1)st interval can be written as

$$\tau = \frac{\int_{ih}^{(i+1)h} \lambda(t-ih)e^{-\lambda t}dt}{\int_{ih}^{(i+1)h} \lambda e^{-\lambda t}dt} = \frac{1-(1+\lambda h)e^{-\lambda h}}{\lambda(1-e^{-\lambda h})} = \frac{1}{\lambda} - \frac{h}{e^{\lambda h}-1} \tag{4}$$

A production cycle has five periods: (i) the in-control period, $1/\lambda + (1-\varepsilon_1)sZ_0\alpha$, (ii) time to generate an out-of-control signal, $h/(1-\beta) - \tau$, (iii) the time to take a sample and analyze the results, gn, (iv) the time to find the assignable cause, Z_1 and (v) the time to eliminate the assignable cause, Z_2. See details in [20]. By considering all these five components, the expected cycle time can be written as follows:

$$E(T) = \frac{1}{\lambda} + (1-\varepsilon_1)sZ_0\alpha + \frac{h}{(1-\beta)} - \tau + gn + Z_1 + Z_2 \tag{5}$$

Where ε_1 is an indicator variable

$$= \begin{cases} 1 & \text{when production continues during search for assignable cause} \\ 0 & \text{when production ceases during search for assignable cause} \end{cases}$$

Z_0=expected search time for a false alarm signal, and
g = time to sample and chart one item.

The cost model consists of the fixed cost a and variable cost b for sampling, as well as following costs

- The expected cost when the process is in-control Q_0/λ
- The expected cost when the process is out-of-control $Q_1/(1-\beta) - \tau + gn + \varepsilon_1 Z_1 + \varepsilon_2 Z_2$
- The expected cost during search period due to a false alarm $sY\alpha$,
- The expected cost for search and repair of true alarm W
- The expected cost due to fixed and variable cost of sampling, (a + bn)[$\{1/\lambda + h/(1-\beta) - \tau + +gn + \varepsilon_1 Z_1 + \varepsilon_2 Z_2\}/h$].

Thus the total quality cost per cycle can be written as follows:

$$E(C) = \frac{Q_0}{\lambda} + Q_1\left[\frac{h}{(1-\beta)} - \tau + gn + \varepsilon_1 Z_1 + \varepsilon_2 Z_2\right] + sY\alpha + W + (a+bn) \times \left[\frac{\{1/\lambda + h/(1-\beta) - \tau + gn + \varepsilon_1 Z_1 + \varepsilon_2 Z_2}{h}\right] \tag{6}$$

Where

Q_0= quality cost per hour while producing in-control,

Q_1= quality cost per hour while producing out-of-control

ε_2 is an indicator variable

$$= \begin{cases} 1 & \text{when production continues during repair} \\ 0 & \text{when production ceases during repair} \end{cases}$$

Y = cost per false alarm, and

W = cost to locate and repair the assignable cause.

Finally, the expected loss cost per unit time can be written as follows:

$$E(L) = \frac{E(C)}{E(T)} \tag{7}$$

The cost function $E(L)$ has three decision variables that are the sample size n, the sampling frequency h, and the control limit width parameter k. In the economic design of the X-bar control chart, this function should be minimized. Thus, it is an unconstrained optimization problem having a discrete-continuous, nonlinear, nondifferentiable objective function with bound constraints of $1 \le n \le 20$, $0.1 \le h \le 5.0$, $0.1 \le k \le 5.0$.

III. DE ALGORITHM

The traditional DE algorithm [21] begins with a population of NP individuals. Each individual in NP has a D-dimensional vector with parameter values. Each vector is obtained randomly and uniformly within the search ranges $[x_{ij}^{min}, x_{ij}^{max}]$ as follows:

$$x_{ij}^{0} = x_{ij}^{min} + \left(x_{ij}^{max} - x_{ij}^{min}\right) \times r \tag{8}$$

where x_{ij}^{g} is the i^{th} target individual with respect to j^{th} dimension at generation g; and r is a uniform random number in [0,1]. Note that for each individual in the population, we keep the following information: $f(x)$ is the objective function value;

In each generation, mutation and crossover operators with parameters F and C_r are applied to each individual x_i ($i = 1,..,NP$). First a mutant individual v_i and then, a trial individual u_i is generated. If $f(u_i)$ is better than $f(x_i)$, x_i will be replaced by u_i. The algorithm evolves the population until the termination criterion (TC) is achieved, and then the best solution of the population is reported. The pseudo-code of DE is shown in Fig. 1.

Step 1. Determine parameters: NP, F, C_r and TC

Step 2. Initialization: Randomly generate a population $NP = \{x_1,..,x_n\}$ and evaluate each solution in NP.

Step 3. Population update: For each individual

a. Perform a mutation operator on x_i to generate v_i.
b. Perform a crossover operator on v_i and x_i to generate a trial vector u_i.
c. Update target individual replace x_i with u_i if $f(u_i) \leq f(x_i)$

Step 4: Termination: If stopping criterion is satisfied, report the best solution x_{best} in NP. Otherwise, go to Step 3.

Fig. 1. Differential evolution Algorithm

Some traditional mutation strategies are presented in the literature as follows:

S1. DE/rand/1:

$$v_i^g = x_a^{g-1} + F \times (x_b^{g-1} - x_c^{g-1})$$

S2. DE/current to best/1:

$$v_i^g = x_i^{g-1} + F\left(x_{best}^{g-1} - x_i^{g-1}\right) + F(x_a^{g-1} - x_b^{g-1})$$

S3. DE/Current to pbest/1:

$$v_i^g = x_i^{g-1} + F\left(x_{pbest}^{g-1} - x_i^{g-1}\right) + F(x_a^{g-1} - x_b^{g-1})$$

S4. DE/Best from Best to current/1

$$v_i^g = x_{best}^{g-1} + U(-1,1) \times (x_{best}^{g-1} - x_i^{g-1})$$

Where a, b, c are three randomly chosen individuals from the target population such that ($a \neq b \neq c \in (1,..,NP)$) and $j = 1,..,D$. $F \in (0,2)$ is a mutation scale factor affecting the differential variation between the two individuals from the population. Note that x_{pbest}^{g-1} is selected by tournament selection with size 2 and $U(-1,1)$ is a uniform random number between -1 and 1. Note that DE/Best to current/1 is presented for the first time in this paper. In addition to the above, note that these mutation strategies are used in the DE-QL algorithm.

During the generation of mutant individuals, they might be outside the search ranges. For this reason, parameter values violating the search range are restricted to below in this paper:

$$v_i^g = \begin{cases} x_i^{min} & if\ v_i^g < x_i^{min} \\ x_i^{max} & if\ v_i^g > x_i^{min} \end{cases}, \tag{9}$$

Trial individuals are obtained by recombining mutant individuals with their corresponding target individuals as follows:

$$u_i^g = \begin{cases} v_i^g & if\ r_i^g \leq C_r\ or\ j = D_j \\ x_i^{g-1} & otherwise \end{cases} \tag{10}$$

where the index D_j is a randomly chosen dimension ($j = 1,..,D$). It makes sure that at least one parameter of the trial individual u_i^g will be different from the target individual x_i^{g-1}. C_r is a user-defined crossover constant in the range [0,1], and r_i^g is a uniform random number in [0,1]. A comprehensive review of DE algorithms can be found in [22-24]

IV. DE_QL ALGORITHM

A. Q-Learning Procedure

In the DE_QL algorithm, the mutation strategy, as well as mutation and crossover rates, are all determined by the Q-Learning algorithm. The Q-learning (QL) is one of the widely used reinforcement learning algorithms. The QL aims to choose an appropriate action based on experience by interacting with the environment. Once the agent (learner) performs a chosen action, it obtains a reward or penalty. Then, it learns to choose the best action to perform by assessing the action alternatives using the cumulative rewards (Q-values).

The Q-values can be calculated for each state-action pair by a Q-learning function given in Eq. (10). Then, Q-values are kept for all state-action pairs in a $Q(s_t, a_t)$ table. Let $S = [s_1, s_2, ... s_p]$ be the set of states, $A = [a_1, a_2, ... a_p]$ be the set of actions, r_{t+1} be the reward, $lf \in [0,1]$ be the learning factor, $df \in [0,1]$ be the discount factor and $Q(s_t, a_t)$ be the Q-value at time *t*. The learner aims to maximize its total reward [25].

$$Q_{t+1}(s_t, a_t) = Q(s_t, a_t) + lf[r_{t+1} + df * max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \tag{11}$$

In the DE-QL algorithm, we assume that there is only one state for each parameter, where the reward value of 1 is given to an action that improves the current solution. We determine the mutation strategy (S), crossover rate (C_r), and mutation rate (F) from the $Q(s_1, a_t)$. Briefly, at each function evaluation, we update the action values in the $Q(s_1, a_t)$ for successful actions. In other words, if a target individual is improved by a chosen action list, a reward 1 is assigned to each of the chosen action list and $Q(s_1, a_t)$ is updated. Then, in the next function evaluation, the algorithm chooses the best action (value/strategy) for each parameter with the maximum action. Note that, in the DE-QL algorithm, we also choose the actions of the parameters randomly with a small jumping probability ($jP = 0.02$) to escape from local minima. The action list of each parameter is given in Table I.

TABLE I. ACTION LIST OF THE PARAMETERS

Parameters	Action List
S	{1, 2, 3, 4}
F	{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}
C_r	{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}

B. Linear Population Reduction Strategy

To improve the performance of DE_QL, a population size reduction strategy is used as in the original LSHADE. The population size NP dynamically decreases with the increasing of FEs according to Eq (12)

$$NP = Round\left(NP_{max} - \frac{Current\ CPU\ Time}{MaxCPU\ Time}(NP_{max} - NP_{min})\right) \quad (12)$$

C. Strategy and Parameter Selection

As mentioned before, mutation strategy, crossover, and mutation rates are selected from the action lists maximizing the QT, but the random selection is also carried out to escape from local minima. Ultimately, the outline of the DE-QL algorithm is given in Fig. 2

Step 1. Determine parameters
Step 2. Initialize population
Step 3. Evaluate initial population
  a. $f(x_1^0), \ldots, f(x_{NP}^0)$
  b. Determine x_{best}
Step 4. Population update: For each x_i^{g-1}
  a. Choose a mutation strategy from $Q(s_1, a_t)$ with jP
  b. Choose the mutation rate from $Q(s_1, a_t)$ with jP
  c. Choose the crossover rate from $Q(s_1, a_t)$ with jP
  d. Generate trial individual u_i^g
  e. Make a selection
     If $\left(f(u_i^g) \leq f(x_i^g)\right)$
     Then $x_i^g = u_i^g$
        R=1 and Update $Q(s_1, a_t)$
     Else
        Then $x_i^g = u_i^g$
  f. Update x_{best}
Step 5. Update population size NP
Step 6. Termination: If the termination criterion is satisfied, report x_{best}. Otherwise, go to step 4.

Note that we do not use the normal distribution table to generate α and β. Instead, we use the erf() function to estimate the α and β. Then, we comupte the objective function value E(L). The fitness function we employed in Fig. 3.

Fig. 2. Outline of DE-QL algorithm

```
int main(){
// constants
double x, y, t, Val, ValB;
double a1 = 0.254829592;
double a2 = -0.284496736;
double a3 = 1.421413741;
double a4 = -1.453152027;
double a5 = 1.061405429;
double p = 0.3275911;
//Parameters
int sign;
int n = Decision variable;
double k = Decision variable;
double h = Decision variable;
double Alpha, Beta, dBeta, Tau, S, ET, EC, EC1, EC2, EL;
double Teta = 0.01;
double Shift = 1.0;
double g = 0.05;//sampling time
double A = 0.5;//fixed cost
double B = 0.10;//SampleVarCost
double Y = 50.0;// FalseAlarmCost
double W = 25.0;//Repair cost
double Q0 = 10;//incontrol quality cost
double Q1 = 100; //out of control Quality cost
double Z0 = 0.0; //false alarm search time
double Z1 = 2.0;//assignable search time
double Z2 = 0.0;//repair time
double IV1 = 1.0;//Indicator variable
double IV2 = 1.0;//Indicator variable
// Save the sign of x
x = -k;
sign = 1;
if (x < 0) sign = -1;
x = fabs(x) / sqrt(2.0);
// A&S formula 7.1.26
t = 1.0 / (1.0 + p*x);
y = 1.0 - (((((a5*t + a4)*t) + a3)*t + a2)*t + a1)*t*exp(-x*x);
Alpha = 2.0*(0.5*(1.0 + sign*y));
printf("%.10f\t", Alpha); system("pause");
x = k - 1.0*sqrt(n);
sign = 1;
if (x < 0) sign = -1;
x = fabs(x) / sqrt(2.0);
// A&S formula 7.1.26
t = 1.0 / (1.0 + p*x);
y = 1.0 - (((((a5*t + a4)*t) + a3)*t + a2)*t + a1)*t*exp(-x*x);
Val = 0.5*(1.0 + sign*y);
x = -k - Shift*sqrt(n);
sign = 1;
if (x < 0) sign = -1;
x = fabs(x) / sqrt(2.0);
// A&S formula 7.1.26
t = 1.0 / (1.0 + p*x);
y = 1.0 - (((((a5*t + a4)*t) + a3)*t + a2)*t + a1)*t*exp(-x*x);
ValB = 0.5*(1.0 + sign*y);
Beta = Val - ValB;
printf("%.10f\t", Beta); system("pause");
dBeta = 1.0 - Beta;
//calculate Tau
Tau = 1.0 / Teta - h / (exp(h*Teta) - 1.0);
printf("%.4f\t", Tau); system("pause");
```

```
//Calcualte S
S = 1.0 / (exp(h*Teta) - 1.0);
printf("%.4f\t", S); system("pause");
//Calculate ET
ET = 1.0 / Teta + (1.0 - IV1)*S*Z0*Alpha + h / (dBeta)-Tau + g*n
+ Z1 + Z2;
printf("%.4f\t", ET); system("pause");
//Calculate EC1
EC1 = Q0 / Teta + Q1*(h / dBeta - Tau + g*n + IV1*Z1 + IV2*Z2)
+ S*Y*Alpha + W;
printf("%.4f\t", EC1); system("pause");
EC2 = (A + B*n)*((1.0 / Teta + h / dBeta - Tau + g*n + IV1*Z1 +
IV2*Z2)) / h;
printf("%.4f\t", EC2); system("pause");
EL = (EC1 + EC2) / ET;
printf("%.4f\t", EL); system("pause");}
```

Fig. 3. Objective function computation

V. COMPUTATIONAL RESULTS

A numerical example in van Deventer and Manna [7] for designing an X-bar control chart has been selected for the present work where the data given are: $\lambda = 0.01$, $\delta = 1$, $g = 0.05$, $a = 0.50$, $b = 0.10$, $Y = 50$, $W = 25$, $Q_0 = 10$, $Q_1 = 100$, $Z_0 = 0$, $Z_1 = 2$, $Z_2 = 0$, $\epsilon_1 = 1$ and $\epsilon_2 = 1$. In this work,we minimize the loss cost function E(L) by the DE_QL algorithm. Regarding three decision variables, sample size n is rounded to integer value during optimization whereas the other two, i.e., sampling interval (h) and width of control limits (k) are taken as real values on a continuous scale, of course restricted to bound constraints.

The initial maximum population size NPmax=100 is established randomly with bound constraints. We carry out 10 replications for the problem on hand and we run the DE_QL algorithm for 2.5 seconds. Population sized is reducted as per generation until a minimum population size minNP=5. Q_L parameters are taken as 1 simply. In each replication, the DE_QL algorithm was able to find the optimal solution. The optimum values of h and k along with corresponding minimum values of expected loss cost E(L) obtained for all integer values of n varying from 1 to 20 have been listed in TABLE II. This table also presents the corresponding values of the two errors α and β. As shown in this table, the optimum value of cost E(L) decreases as n value increases from 1 to 12 and then increases at higher values of n. This trend is also graphically shown in Fig. 4. The rate of reduction of loss cost E(L)is observed to be very large as the sample size increases in the beginning and then the rate gradually diminishes till the cost becomes minimum. Thus, the minimum possible cost is found to be E(L) = 10.8376 and this occurs at n = 12. The corresponding values of h and k at an optimum solution are 1.8464 and 2.6198, respectively. The same trend of the relationship between E(L) and n has also been reported by van Deventer and Manna [7] as shown in TABLE II. They have also obtained the most minimum cost at n = 12. On comparison of results, it is clear that at all values of sample size; the optimum costs obtained by the DE_QL algorithm are lower than that of van Deventer andManna [7]. This might be due to the rounding error in the objective function calculation by erf() function.

TABLE II. ECONOMIC DESIGN RESULTS

	DE_QL					Van Deveventer and Manna					TLBO				
n	h	k	alpha	beta	E(L)	h	k	alpha	beta	E(L)	h	k	alpha	beta	E(L)
1	0.5962	2.1569	0.0310	0.8755	19.2018	0.7	2.1	0.3570	0.8634	19.2200	0.5969	2.1562	0.0319	0.8734	19.2036
2	0.6851	2.2860	0.0223	0.8082	17.3503	0.7	2.3	0.0214	0.8120	17.3600	0.6856	2.2854	0.0222	0.8086	17.3518
3	0.8092	2.3472	0.0189	0.7308	16.4208	0.9	2.3	0.0214	0.7149	16.4300	0.8104	2.3471	0.0193	0.7284	16. 4219
4	0.9422	2.3884	0.0169	0.6511	15.8687	0.9	2.4	0.0164	0.6554	15.8700	0.9424	2.3879	0.0174	0.6472	15.8697
5	1.0742	2.4203	0.0155	0.5731	15.5121	1.1	2.4	0.0164	0.5651	15.5100	1.0746	2.4202	0.0156	0.5726	15.5131
6	1.2038	2.4485	0.0143	0.4996	15.2716	1.3	2.4	0.0164	0.4803	15.2800	1.2044	2.4484	0.0135	0.5084	15.2725
7	1.3268	2.4763	0.0133	0.4327	15.1064	1.3	2.5	0.0124	0.4421	15.1100	1.3277	2.4763	0.0134	0.4318	15.1071
8	1.4453	2.5038	0.0123	0.3727	14.9933	1.5	2.5	0.0124	0.3713	14.9900	1.4453	2.5037	0.0128	0.3678	14.9940
9	1.5553	2.5315	0.0114	0.3197	14.9179	1.6	2.5	0.0124	0.3085	14.9200	1.5560	2.5314	0.0127	0.3063	14.9185
10	1.6597	2.5602	0.0105	0.2736	14.8708	1.6	2.6	0.0093	0.2870	14.8700	1.6597	2.5600	0.0105	0.2736	14.8714
11	1.7565	2.5892	0.0096	0.2335	14.8456	1.7	2.6	0.0093	0.2368	14.8500	1.7567	2.5894	0.0096	0.2336	14.8461
12	**1.8464**	**2.6198**	**0.0088**	**0.1993**	**14.8376**	1.9	2.6	0.0093	0.1938	14.8400	1.8474	2.6196	0.0088	0.1991	14.8381
13	1.9317	2.6502	0.0080	0.1697	14.8434	1.9	2.7	0.0069	0.1826	14.8500	1.9324	2.6505	0.0080	0.1701	14.8438
14	2.0121	2.6816	0.0073	0.1446	14.8605	2.0	2.7	0.0069	0.1488	14.8600	2.0122	2.6819	0.0073	0.1446	14.8608
15	2.0869	2.7138	0.0067	0.1232	14.8867	2.1	2.7	0.0069	0.1204	14.8900	2.0873	2.7139	0.0067	0.1232	14.8870
16	2.1584	2.7462	0.0060	0.1050	14.9204	2.2	2.7	0.0069	0.0968	14.9200	2.1584	2.7463	0.0061	0.1048	14.9207
17	2.2260	2.7785	0.0055	0.0894	14.9604	2.2	2.8	0.0051	0.0929	14.9600	2.2258	2.7790	0.0054	0.0898	14.9607
18	2.2895	2.8118	0.0049	0.0762	15.0056	2.3	2.8	0.0051	0.0746	15.0100	2.2899	2.8120	0.0051	0.0748	15.0059
19	2.3509	2.8448	0.0044	0.0650	15.0552	2.4	2.8	0.0051	0.0595	15.0600	2.3512	2.8453	0.0046	0.0640	15.0555
20	2.4101	2.8782	0.0040	0.0555	15.1084	2.4	2.9	0.0037	0.0580	15.1100	2.4102	2.8785	0.0040	0.0552	15.1087

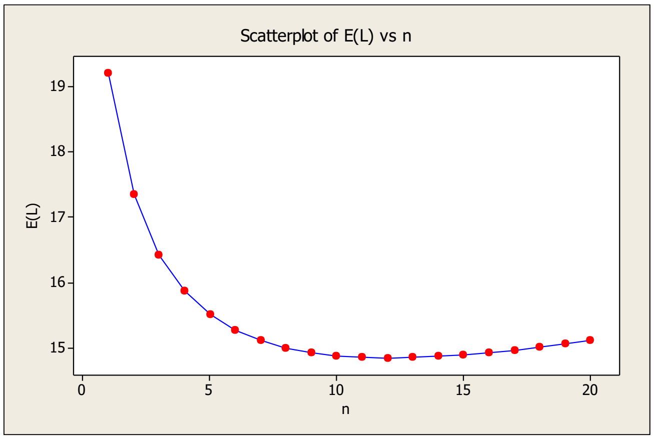

Fig. 4. Scatter plot of E(L) vs n

The mathematical model of the X-bar chart has nine factors with two-level cost parameters as shown in Table IV. The low and high values of all the nine parameters considered by Chen and Tirupati [26] and Ganguly and Patel [20] are also given in Table IV. Since each factor has two-level, we employ a $2k - p$ fractional factorial design of resolution IV to analyze the impact of the objective function parameters on n, h, k, and $E(L)$. A 2_{IV}^{9-4} fractional factorial design with 32 runs is chosen for the model. We run the DE_QL algorithm for each of 32 treatments and list the values of design parameters n, h, k, and $E(L)$, respectively on Table V.

TABLE III. FACTORS AND THEIR LEVELS IN THE DOE

		Level	
Num. of Par.	Factors	Low (-1)	High (+1)
1	Lamda	0.01	0.05
2	Shift	1	2
3	A	0.50	5
4	B	0.10	1
5	Y	50	500
6	W	25	50
7	Q0	10	50
8	Q1	50	1000
9	G	0.00	0.05

TABLE IV. DOE RESULTS

Design								n	h	k	E(L)
1	-1	1	1	1	-1	1	-1	1	4.9997	4.9840	51.2002
-1	-1	1	1	-1	1	1	-1	1	4.9999	4.8062	60.2007
1	-1	1	1	-1	1	-1	1	12	1.8165	2.2283	206.9663
-1	-1	-1	1	-1	-1	-1	-1	11	4.9973	2.7798	15.6936
1	1	-1	-1	1	1	1	-1	1	4.9999	4.8415	60.0206
1	-1	-1	-1	-1	1	-1	-1	19	4.9982	2.3512	29.1625
1	-1	-1	1	-1	-1	1	1	16	0.8578	2.7515	180.0285
-1	-1	-1	-1	-1	1	1	1	12	3.4228	1.6107	106.9321
-1	1	1	-1	-1	1	-1	-1	16	5.0000	2.5133	22.8200
1	1	-1	1	-1	1	1	-1	1	5.0000	4.9605	60.2002
1	-1	1	-1	-1	-1	1	-1	2	4.9996	4.9438	51.0203
-1	1	1	1	1	1	-1	-1	12	5.0000	2.8341	25.4064
1	-1	-1	-1	1	-1	1	1	5	0.1587	2.3901	168.8732
-1	-1	-1	-1	1	-1	-1	-1	17	3.3322	2.7781	12.5174
-1	-1	1	-1	1	1	1	-1	2	5.0000	4.6713	60.0206
-1	1	-1	1	-1	1	-1	1	20	3.8380	2.6096	68.2466
-1	-1	1	1	1	-1	-1	1	12	1.7613	2.6146	59.6329
-1	-1	1	-1	-1	-1	-1	1	19	1.1987	2.6060	41.9298
-1	1	-1	1	1	-1	1	-1	1	4.9990	4.9477	50.3003
1	-1	-1	1	1	1	-1	-1	14	4.9983	2.5969	33.6355
-1	1	1	1	-1	-1	1	1	13	1.7919	2.6226	98.1192
-1	1	1	-1	1	-1	1	1	19	1.2228	2.6108	80.8942
-1	1	-1	-1	1	1	-1	1	11	3.3561	1.6093	68.6214
1	-1	1	-1	1	1	-1	1	20	1.6456	2.1708	167.0071
1	1	-1	-1	-1	-1	-1	1	5	0.1625	2.3866	133.3722
1	1	1	1	-1	-1	-1	-1	16	4.9953	2.6838	23.8362
-1	1	-1	-1	-1	-1	1	-1	1	4.9997	4.9983	50.1202
1	1	1	1	1	1	1	1	13	1.8579	2.2248	240.6472
1	1	-1	1	1	-1	-1	1	16	0.8411	2.7680	144.5664
1	1	1	-1	-1	1	1	1	19	1.6720	2.1352	202.1654
1	1	1	-1	1	-1	-1	-1	14	3.0412	2.3263	20.7435
-1	-1	-1	1	1	1	1	1	20	4.0120	2.7419	106.6465

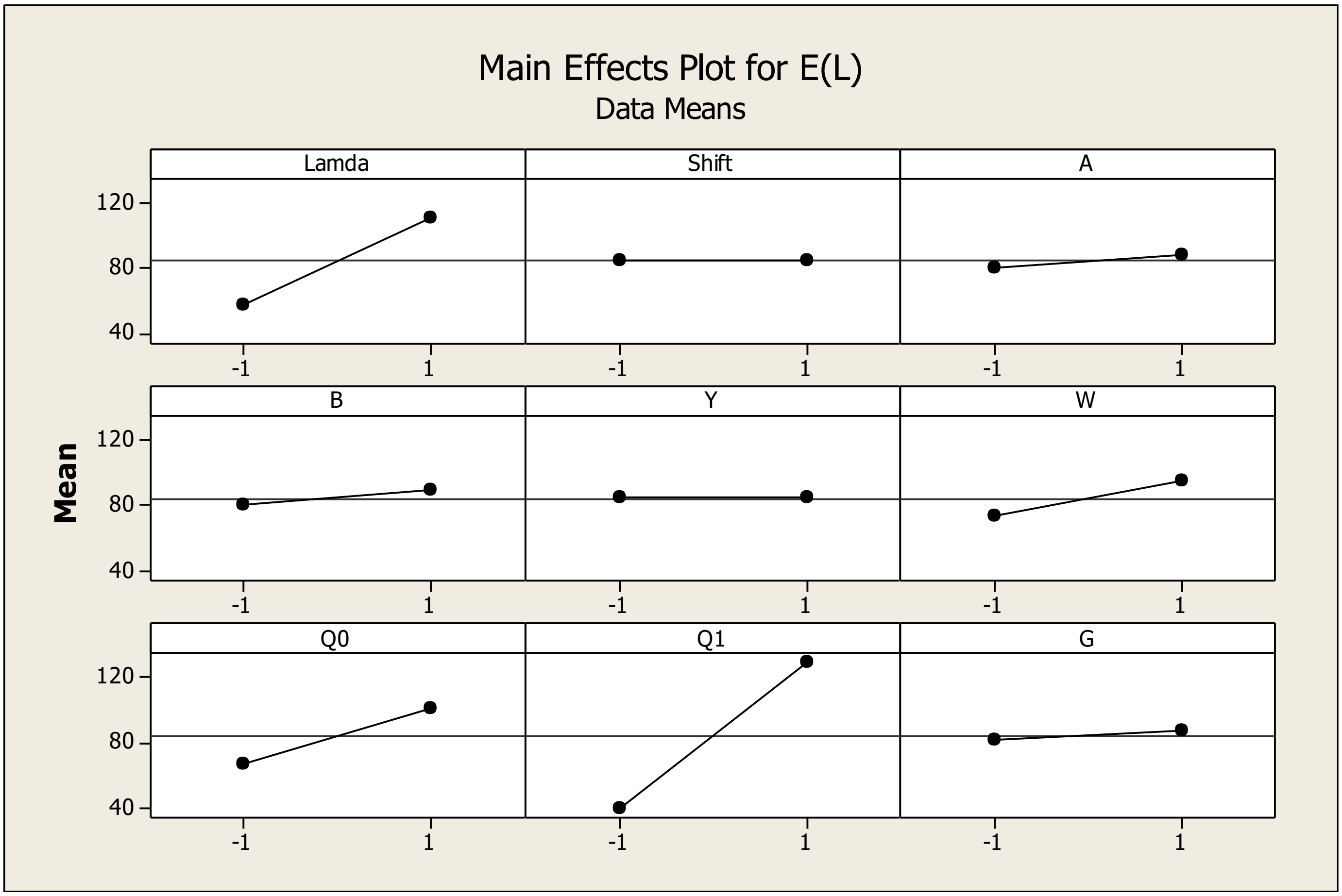

Fig. 5. Main effects plot of parameters

We provide the main effects plot of response variable $E(L)$ in Fig. 5. To minimize the $E(L)$, Lamda, Q0, Q1, and W should be chosen as low values as shown in Fig. 5. Shift, A, B, Y, and G could be either low or high values.

VI. Conclusion

This paper presents a novel differential evolution algorithm with Q-Learning (DE-QL) for the economical and statistical design of X-Bar control charts, which has been commonly used in industry to control manufacturing processes. In the X-Bar charts, samples are collected from the production process at regular intervals in order to measure a quality characteristic and the sample means are plotted on this chart. When designing a control chart, three parameters should be selected, namely, the sample size (n), the sampling interval (h), and the width of control limits (k). On the other hand, when designing an economical and statistical design, these three control chart parameters should be selected in such a way that the total cost of controlling the process should be minimized by finding optimal values of these three parameters. In this paper, we develop a DE_QL algorithm for the global minimization of a loss cost function expressed as a function of three variables n, h, and k in an economic model of the X-bar chart. A problem instance that is commonly used in the literature has been solved and competitive and or slightly better results are found than the earlier published results.

References

[1] D.C. Montgomery, Introduction to Statistical Quality Control, 7th ed., JohnWiley and Sons, Inc., New York, 2013.

[2] D.C. Montgomery, The economic design of control charts: a review and literature survey, J. Qual. Technol. 12 (1980) 75–87.

[3] A.J. Duncan, The economic design of ¯Xcharts used to maintain current control of a process, J. Am. Stat. Assoc. 51 (1956) 228–242.

[4] M.R. Panagos, R.G. Heikes, D.C. Montgomery, Economic design of X-bar control charts for two manufacturing process models, Nav. Res. Logist. Q. 32 (1985)631–646.

[5] L. Svoboda, Economic design of control charts: a review and literature survey1979–1989, in J.B. Keats, D.C. Montgomery (Eds.), Statistical Process Control in Manufacturing, Marcel Dekker, New York, 1991, pp. 311–330.

[6] C. Ho, K.E. Case, Economic design of control charts: a literature review for1981–1991, J. Qual. Technol. 26 (1994) 39–53.

[7] P.J.U. van Deventer, Z.G. Manna, An easy and low-cost option for economic statistical process control using Excel, Orion 25 (2009) 1–15.

[8] F.J. Yu, C.S. Tsou, K.I. Huang, Z. Wu, An economic-statistical design of ¯Xcontrolcharts with multiple assignable causes, J. Qual. 17 (2010) 327–338.

[9] M. Chiha, L.L. Yeh, F.C. Lic, Particle swarm optimization for the economic and economic statistical designs of the ¯X control chart, Appl. Soft Comput. 11 (2011)5053–5067.

[10] S.T.A. Niaki, M. Malakib, M.J. Ershadi, A particle swarm optimization approach on economic and economic-statistical designs of MEWMA control charts, Sci.Iran. 18 (2011) 1529–1536.

[11] W. Yang, Y. Guo, W. Liao, Economic and statistical design $\bar{X}$and S control charts using an improved multi-objective particle swarm optimization algorithm, Int.J. Prod. Res. 50 (2012) 97–117.

[12] S.T.A. Niaki, M.J. Ershadi, A hybrid ant colony, Markov chain, and experimental design approach for the statistically constrained economic design of MEWMAcontrol charts, Expert Syst. Appl. 39 (2012) 3265–3275.

[13] M.A.A. Cox, An approximate approach to the economic design of $\bar{X}$ charts by considering the cost of quality, J. Mod. Appl. Stat. Methods 12 (2013) 170–180.

[14] T.J. Lorenzen, L.C. Vance, The economic design of control charts: a unified approach, Technometrics 28 (1986) 3–10.

[15] V.B. Vijaya, S.S.N. Murthy, A new approach to the robust economic design of control charts, Appl. Soft Comput. 7 (2007) 211–228.

[16] G. Gupta, S.K. Patel, Economic design of $\bar{X}$– control chart using particle swarm optimization, Int. J. Adv. Manuf. Syst. 2 (2011) 29–34.

[17] M. Chih, L.L. Yeh, F.C. Li, Particle swarm optimization for the economic and economic statistical designs of the $\bar{X}$control chart, Appl. Soft Comput. 11 (2011)5053–5067.

[18] A. Ganguly, S.K. Patel, Economic design of $\bar{X}$control chart using simulated annealing, Procedia Eng. 38 (2012) 1037–1043.

[19] S.T.A. Niaki, M. Malakib, M.J. Ershadi, A particle swarm optimization approach on economic and economic-statistical designs of MEWMA control charts, Sci.Iran. (2011) 1–8, http://dx.doi.org/10.1016/j.scient.2011.09.007.

[20] A. Ganguly, S. K. Patel A teaching-learning based optimization approach for the economic design of X-bar control chart, Applied Soft Computing 24 (2014) 643–653.

[21] R. Storn and K. Price, “Differential Evolution - A simple evolution strategy for fast optimization,” Dr. Dobb’s J., vol. 22, no. 4, pp. 18–24, 1997.

[22] S. Das and P. N. Suganthan, “Differential evolution: A survey of the state-of-the-art,” IEEE Trans. Evol. Comput., vol. 15, no. 1, pp. 4–31, 2011.

[23] S. Das, S. S. Mullick, and P. N. Suganthan, “Recent advances in differential evolution-An updated survey,” Swarm Evol. Comput., vol. 27, pp. 1–30, 2016.

[24] S. Das, S. S. Mullick, and P. N. Suganthan, “Recent advances in differential evolution-An updated survey,” *Swarm Evol. Comput.*, vol. 27, pp. 1–30, 2016.

[25] S.S. Choong, L-P. Wong, and C.P. Lim, “Automatic design of hyper-heuristic based on reinforcement learning,” *Information Sciences*, vol. 436-437, pp. 89-107, 2018.

[26] W.H. Chen, D. Tirupati, Economic design of $\bar{X}$ control charts: insights on designvariables, Int. J. Qual. Reliab. Manag. 14 (1997) 234–259.