Expert Systems With Applications 50 (2016) 17–39

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Multi-objective control chart design optimization using NSGA-III and MOPSO enhanced with DEA and TOPSIS

Madjid Tavana [a,b,*], Zhaojun Li [c], Mohammadsadegh Mobin [c], Mohammad Komaki [d], Ehsan Teymourian [e]

[a] Business Systems and Analytics Department, Distinguished Chair of Business Analytics, La Salle University, Philadelphia, PA19141, USA.
[b] Business Information Systems Department, Faculty of Business Administration and Economics, University of Paderborn, D-33098 Paderborn, Germany
[c] Industrial Engineering and Engineering Management Department, Western New England University, Springfield, MA01119, USA
[d] Department of Electrical Engineering and Computer Science, Case Western Reserve University, Cleveland, OH44106, USA
[e] School of Mechanical, Industrial, and Manufacturing Engineering, Oregon State University, Corvallis, OR 97331 USA

ARTICLE INFO

Keywords:
Economical control chart design
NSGA-III algorithm
MOPSO algorithm
Data envelopment analysis
TOPSIS

ABSTRACT

X-bar control charts are widely used to monitor and control business and manufacturing processes. This study considers an X-bar control chart design problem with multiple and often conflicting objectives, including the expected time the process remains in statistical control status, the type-I error, and the detection power. An integrated multi-objective algorithm is proposed for optimizing economical control chart design. We applied multi-objective optimization methods founded on the reference-points-based non-dominated sorting genetic algorithm-II (NSGA-III) and a multi-objective particle swarm optimization (MOPSO) algorithm to efficiently solve the optimization problem. Then, two different multiple criteria decision making (MCDM) methods, including data envelopment analysis (DEA) and the technique for order of preference by similarity to ideal solution (TOPSIS), are used to reduce the number of Pareto optimal solutions to a manageable size. Four DEA methods compare the optimal solutions based on relative efficiency, and then the TOPSIS method ranks the efficient optimal solutions. Several metrics are used to compare the performance of the NSGA-III and MOPSO algorithms. In addition, the DEA and TOPSIS methods are used to compare the performance of NSGA-III and MOPSO. A well-known case study is formulated and solved to demonstrate the applicability and exhibit the efficacy of the proposed optimization algorithm. In addition, several numerical examples are developed to compare the NSGA-III and MOPSO algorithms. Results show that NSGA-III performs better in generating efficient optimal solutions.

1. Introduction

Statistical process control (SPC) is one of the most effective continuous quality improvement strategies, which uses different statistical methods to improve quality and productivity in industrial processes. SPC is widely used to monitor and control business and manufacturing processes. A statistical control chart is the primary SPC tool used to differentiate "assignable" sources of variation from "common" sources. The engineering implementation of control charts involves a number of technical and behavioral decision-making processes (such as design) which are very important in the application of control charts. The design of a control chart usually refers to the selection of parameters such as the sample size (n), the control limit width (k) in terms of standard deviation (σ), and the time interval between succeeding samples (sampling frequency) (h). Configuring this design plays an important role in improving the firms' overall quality and productivity. The performance metrics of a control chart design include economics based measures (cost-effectiveness), the speed of detecting mean shifts (effectiveness), and the numebr of false alarms (efficiency). A large number of methods have been proposed for designing control charts including Shewhart's simple rule, statistical criteria, economic criteria, or joint economic and statistical criteria. Each method has some advantages and disadvantages such as complexity in implementation, statistical configurations, and cost effectiveness (Saniga, 1989). Faraz and Saniga (2013) argue that most existing control chart design models lack flexibility and adaptability in real-world problem solving and propose multi-objective models as viable alternatives. The

* Corresponding author at: Business Systems and Analytics Department, Distinguished Chair of Business Analytics, La Salle University, Philadelphia, PA19141, United States. Tel.: +1 215 951 1129; fax: +1 267 295 2854.
E-mail addresses: tavana@lasalle.edu (M. Tavana), zhaojun.li@wne.edu (Z. Li), mm337076@wne.edu (M. Mobin), gxk152@case.edu (M. Komaki), teymoure@oregonstate.edu (E. Teymourian).
URL: http://tavana.us/ (M. Tavana)

http://dx.doi.org/10.1016/j.eswa.2015.11.007
0957-4174/

multi-objective control chart design models produce a set of optimal solutions (instead of a single optimal solution) that allows decision-makers to tailor their solution to a specific industrial situation at a particular time.

Duncan (1956) proposed one of the first economic based design models for control charts. He proposed a single objective formulation for Shewhart's original X-bar control chart and considered a production process with a single assignable cause. Duncan (1971) proposed a new model using multi-assignable causes. Chung (1994) developed an algorithm for computing the economically optimal X-bar control charts for a process with multi-assignable causes. Chen and Yang (2002) studied the economic design of c-bar control charts with Weibull in-control times under multi-assignable causes. Yu and Hou (2006) optimized the control chart parameters with multi-assignable causes and variable sampling intervals.

Several models have been proposed for a number of Shewhart-type charts, e.g., the X-bar and R charts employed jointly by Saniga (1977). Lorenzen and Vance (1986) extended Duncan's model and developed a general model for different control chart configurations. Montgomery (1980) and Ho (1994) have provided a good summary of the economic design literature. In spite of the popularity of the economical control chart design model introduced by Duncan (1956), Woodhall (1986) criticized this model and pointed out that Duncan's model is only based on the total cost and could not be used for decision-making problems with inadequate statistical properties (i.e., a large number of false alarms and a poor shift detection power). He showed the difficulties in estimating some of the costs and argued that optimal economical control charts often have poor statistical performance. Saniga (1989) later proposed the first statistical-economic design by improving the Shewhart-type control charts and applying them to the joint determination of the parameters for the X-bar and R charts. Saniga's (1989) design minimized the economic-loss cost function and considered constraints that included an upper bound on the average time to signal an out of control condition, and a lower bound on the power for some customers' specified shift sizes, as well as a lower bound on the type-I error.

Like many engineering problems, the economical control chart design optimization problem has multiple objectives including: maximizing average run length, maximizing probability of detection, and minimizing cost. There are two general approaches for optimizing multiple objective problems. The first approach is to integrate all the objective functions as a single composite function. This approach sums multiple objective functions with different weights. For economical control chart design, all objective functions including economic-based and statistic-based objectives can be converted into a cost function. However, this conversion is not often practical in real-world problems. In addition, determining an appropriate utility function, such as selecting appropriate weights for each objective, is difficult when combining multiple and often conflicting objectives. The second approach is to select one of the objectives as the primary objective and the remaining objectives as constraints (e.g., see Saniga (1989)). Nevertheless, considering some objectives as constraints by specifying boundary values may reduce the solution space and lose candidate solutions preferable to decision makers. Determining the statistical bounds of the model is also difficult in practice. In summary, both approaches have some practical difficulties in real-world problems.

In addition to the weaknesses pointed out by Woodall (1986), some researchers have pointed out that the economic design of control charts also suffer from the difficulties in estimating the costs in Duncan's (1956) original model. To address this issue, Castillo, Mackin, and Montgomery (1996) proposed a multi-criteria model for optimal control chart design using an interactive multiple criteria decision making (MCDM) algorithm which uses a linear value function to approximate the customers' decision preference. The model involves three criteria including the expected number of false alarms, the average time to signal, and the sampling cost per cycle. It also contains two statistical constraints including an upper bound on the false alarms and a lower bound on the power. Such a formulation avoids the difficulties of estimating the costs associated with false alarms, the operation in out-of-control status, and the investigation of assignable causes. However, the linear approximation of the value function may not represent the utility function of the decision maker. Furthermore, experts may have difficulties in determining the statistical bounds of the model. Celano and Ficher (1999) proposed a multi-objective decision making (MODM) approach based on the evolutionary algorithm (EV) to optimize the type-I and type-II error probabilities, as well as the cost of X-bar control chart.

More recently, multi-objective control chart design optimization problems have been extensively investigated in the literature. Chen and Liao (2004) considered all possible combinations of design parameters as a solutions set. Each solution represents a decision making unit (DMU), characterized by three criteria including the average hourly cost, the average run length of process when it is in control status, and the detection power of the chart, with the same constraints presented by Saniga (1989). Data envelopment analysis (DEA) was used to compare the DMUs based on the relative efficiency factor. Li, Kapur, and Chen (2009) analyzed the design of the X-bar control chart problem using a DEA-based multi-criteria branch and bound algorithm. They utilized measurable costs in their model. The immeasurable costs and the long term quality performance indices were expressed by the higher average run length (ARL_0) and the detection power (p). Asadzadeh and Khoshhalan (2009) applied DEA to obtain the optimal design in control chart design problems when the cost function is extended from a single to multi-assignable causes. Faraz and Saniga (2013) optimized a two-objective economic statistical control chart design problem (X-bar and S-squared charts) using genetic algorithm. They also solved a well-known industrial problem at a General Motors casting operation described by Lorenzen and Vance (1986) and compared optimal multi-objective designs with economic designs, statistical designs, and economic statistical designs. Their comparison showed that multi-objective design is more applicable and effective in a real-word situation.

Safaei, Kazemzadeh, and Niaki (2012) merged the Taguchi lost function and the intangible external costs in the economic design of an X-bar control chart and used the non-dominated sorting genetic algorithm (NSGA-II) to find the Pareto optimal solutions. Yang, Guo, and Liao (2012) applied a multi-objective particle swarm optimization (MOPSO) algorithm to find the optimal design of X-bar and S control charts. Recently, Mobin, Li, and Massahi Khoraskani (2015) generated the Pareto optimal frontier of an X-bar control chart problem using the NSGA-II algorithm. They subsequently reduced the Pareto optimal solutions to a managable size.

A review of the multi-objective control chart design literature shows that there are three different groups of optimization methods. In the first group, DEA tools are used to compare all the combinations of the design parameters as DMUs. The fundamental shortcoming of this method is not considering all the potential designs or feasible solutions as DMUs. The second group is used most commonly and includes optimization algorithms (i.e., metaheuristics) to generate the optimal designs. However, these methods also have a shortfall in selecting the best design due to a large number of optimal solutions. In the third group, tools such as DEA are used to find the efficient solutions from all of the optimal solutions generated by the optimization algorithm. These hybrid methods also suffer from a similar drawback in spite of the reduced number of solutions.

We propose two metaheuristic algorithms, a multi-objective optimization method based on the NSGA-II algorithm, called NSGA-III, and a MOPSO algorithm, and integrate them with DEA and the technique for order of preference by similarity to ideal solution (TOPSIS) in order to address many of the deficiencies of the traditional approaches in the literature for solving the X-bar chart design problem.

Table 1
Nomenclature used in the multi-objective X-bar economical control chart model.

Notation	Description
n	sample size
h	The time interval between successive samples (sampling frequency)
k	The control-limit width in terms of standard deviations σ
$s = (n, h, k)$	The DMU vector
$ARL_0(s)$	Average run length (ARL) with the process in control state $(1/\alpha)$, affected by $s = (n, h, k)$
$p(s)$	The detection power of the economical control chart (after out-of-control status), affected by $s = (n, h, k)$
p_L	The lower bound of the economical control chart detection power
τ	The expected time of occurrence of the assignable cause within the interval between two samples
D	The time to search the assignable cause and make the process work at the in-control state again
λ	Assignable cause follows a Poisson process with rate λ
$1/p$	The expected number of samples taken before detecting a mean shift of the process
$\Phi(z)$	The probability density function of the standardized normal distribution
δ	The number of standard deviations σ in the shift of process mean μ_0
$\alpha(s)$	The probability of false alarm (type-I error), which is affected by $s = (n, h, k)$
α_U	The upper bound of the type-I error
$E_{CT}(s)$	The average cycle length
$E_{CC}(s)$	The expected cost per cycle
$E_{HC}(s)$	The expected cost per hour
a_1	The fixed cost of sampling
a_2	The variable cost of sampling
a_3	The cost of searching for an assignable cause
a_4	The cost of investigating a false alarm
a_5	The hourly penalty cost associated with production in the out-of-control state
g	A constant used to estimate the average time of sampling, inspection, evaluation and plotting for each sample (gn)

We also use several metrics to compare the performance of the two metaheuristic algorithms. In addition, we use four different DEA models to compare the solutions in terms of their relative efficiency. Finally, a popular MCDM method called TOPSIS is utilized to rank the optimal efficient solutions. The application of the integrated optimization method produces a set of prioritized optimal efficient designs for implementation.

The remainder of this paper is organized as follows: Section 2 provides a mathematical model and describes the assumptions for the multi-objective X-bar economical control chart design problem. Section 3 presents two optimization methods (i.e., NSGA-III and MOPSO) modified and utilized to solve the control chart design problem. Section 4 presents a MCDM model (i.e., DEA) for reducing the number of optimal solutions and a second MCDM model (i.e., TOPSIS) to rank the optimal solutions. In Section 5 we present a numerical example (solved by the NSGA-III and MOPSO algorithms) to show the applicability of the proposed models and exhibit the efficacy of the procedures and algorithms with respect to several metrics. In addition, in this section, we demonstrate the application of four different DEA models for finding the efficient solutions, and the application of TOPSIS for ranking the efficient solutions. We conclude in Section 6 with our conclusion and future research directions.

2. Multi-objective X-bar economical control chart design

All notations used in describing the economical X-bar control chart design are presented in Table 1.

The main assumptions of an X-bar economical control chart design problem are summarized as follows. The process is assumed to start in statistical control status. It is also assumed that only one type of assignable cause can lead to a fixed shift in the process. An assignable cause is assumed to follow a Poisson process with a rate of λ. In other words, the time interval over which the process remains in in-control-status is an exponential random variable with a mean of $1/\lambda$ hour(s). It is also assumed that the sampling data follows a normal distribution. Once an assignable cause happens, the process mean shifts from μ_0 to $\mu_0 + \delta * \sigma$, with a known δ. The process is not fixed even after an out-of-control (OOC) point is detected, and the time from the detection of an OOC point to the restoration of the process to in-control status is denoted by D.

Considering the above assumptions, some quantitative performance metrics can be derived as follows: (I) The expected time interval that the process continues in statistical control status is $1/\lambda$ due to the Poisson occurrence of an assignable cause; (II) The type-I error or the probability of false alarm ($\alpha(s)$) in each sampling, which is the reciprocal of the average run length with the process in control state, is represented by Eq. (1); (III) The detection power $p(s)$ for each sample after the process is in OCC status is represented by Eq. (2), where $\Phi(z)$ is the standard normal probability density function. The expected number of samples taken before detecting the mean shift is $1/p$; (IV) The conditional expectation time of assignable cause occurring within any two successive samples (given that an assignable cause occurs in this interval) is represented by Eq. (3), where h is the time interval between succeeding samples.

$$\alpha(s) = \frac{1}{ARL_0(s)} = 2\int_{-\infty}^{-k} \Phi(z)dz \tag{1}$$

$$p(s) = \int_{-\infty}^{-k-\delta\sqrt{n}} \Phi(z)dz + \int_{k-\delta\sqrt{n}}^{\infty} \Phi(z)dz \tag{2}$$

$$\tau = \frac{\int_{jh}^{(j+1)h} \lambda e^{-\lambda t}(t - jh)dt}{\int_{jh}^{(j+1)h} \lambda e^{-\lambda t}dt} = \frac{1 - (1 + \lambda h)e^{-\lambda h}}{\lambda\left(-e^{-\lambda h}\right)} \tag{3}$$

In addition to the above-mentioned metrics, there are other performance measures in the literature, such as the expected number of samples when the process is in statistical control status (which can be calculated as $1/\lambda h$, hence the expected number of false alarms during this period is equal to $\alpha(s)/(\lambda h)$). Fig. 1 (Mobin et al., 2015) depicts an entire cycle of the process, including both in-control and out-of-control status.

As shown in Fig. 1, h is the time interval between succeeding samples. The average time spent on sampling, inspecting, evaluating, and plotting each sample is a constant g proportional to the sample size n. Thus, the time delay in this phase is $g * n$. The parameter τ is the expected time of occurrence of the assignable cause within the interval between two samples, and the time to search the assignable cause and make the process return to an in-control status is a constant D.

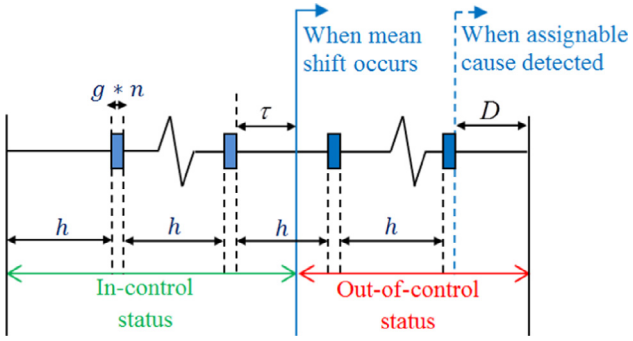


Fig. 1. A single process cycle.

The average cycle length of the process for a design $s = (n, h, k)$ can be expressed by Eq. (4):

$$E_{CT}(s) = \frac{1}{\lambda} + \left(\frac{h}{p(s)} - \tau\right) + gn + D \tag{4}$$

Furthermore, the expected cost per cycle under a design $s = (n, h, k)$ can be expressed by Eq. (5):

$$E_{CC}(s) = \frac{(a_1 + a_2 n)E_{CT}(s)}{h} + a_3 + \frac{a_4 \alpha(s) \exp(-\lambda h)}{1 - \exp(-h\lambda)} + a_5\left(\frac{h}{p(s)} - \tau + gn + D\right) \tag{5}$$

where a_1 is the fixed cost of sampling; a_2 is the variable cost of sampling; a_3 is the cost of searching for an assignable cause; a_4 is the cost of investigating a false alarm; and a_5 is the unit time penalty cost associated with production in the out-of-control status. The cost per time unit (per hour) $E_{HC}(s)$ under a design s can be obtained by dividing $E_{CC}(s)$, the expected cost per cycle, by $E_{CT}(s)$, the average cycle length.

In this paper, three criteria are derived based on the original economic design model of Duncan (1956). By considering two statistical constraints (the upper bound α_U of the type-I error and the lower bound p_L of the detection power), which were integrated by Saniga (1989) into his economic model, the multi-objective X-bar economical control chart design can be formulated as follows:

$$\max_s \ f_1 = ARL_0(s) \tag{6}$$

$$\max_s f_2 = p(s) \tag{7}$$

$$\min_s f_3 = E_{HC}(s) \tag{8}$$

$$\text{s.t.} \quad p(s) \geq p_L \tag{9}$$

$$\alpha(s) \leq \alpha_U, \quad \forall s = (n, h, k) \tag{10}$$

The decision variables in the aforementioned multi-objective economical control chart design are the sample size n, the control limits k, with respect to a known process standard deviation σ, and the sampling frequency of two successive samples within the interval h. One possible design for the control chart consists of a combination of n, h, and k, denoted by $s = (n,\ h,\ k)$. Note that in the above problem, $n \in \mathbb{Z}^+$ is discrete while h and k are continuous variables, i.e., $h, k \in \mathbb{R}^+$. Complex MODM methods, involving software, can be utilized to select the best compromise solutions for the foregoing mathematical model.

In this paper, we propose the integrated framework presented in Fig. 2 to optimize an economical control chart design problem. After defining the multi-objective design of the control chart problem and presenting its mathematical model (Phase 1), the modified NSGA-III and MOPSO algorithms are utilized to generate an optimal design in Pareto frontier format (Phase 2). Next, the efficient optimal designs are obtained using the relative efficiency concept (Phase 3) since the number of optimal designs obtained in Phase 1 is not of a workable size. Four DEA methods are then used to obtain the best designs in terms of relative efficiency. Finally, in Phase 4, the TOPSIS method is utilized to rank the efficient optimal designs.

3. Optimization methods for solving multi-objective X-bar economical control chart design problems

An X-bar economical control chart design problem generally involves multiple and often conflicting objectives such as maximizing average long length when in statistical control, maximizing the detection power of the chart and minimizing the average cost. In traditional optimization methods, there are two general approaches to solve multi-objective problems. The first approach is to integrate all the objective functions into a single composite function and the second approach is to select one of the objectives as a main objective and consider the other objectives as constraints. Both methods may have some difficulties (e.g., determining an appropriate utility function) in practice. In addition, selecting appropriate weights for each objective may become difficult when combining multiple incomparable objectives. Moreoever, considering some objectives as constraints by specifying boundary values may reduce the solution space and lose potential candidate solutions preferable to decision-makers.

On the other hand, evolutionary algorithms (EAs) are considered very effective in solving multi-objective optimization problems by providing a set of non-dominated solutions, known as the Pareto frontier, as the optimal solutions. A wide range of multi-objective EAs have been developed, which can be divided into two categories. The first category, which includes NSGA (Srinivas & Deb, 1994), does not provide a tool for the preservation of good solutions (elitism); whereas the second category, known as NSGA-II (Deb, Agrawal, Pratap, & Meyarivan, 2000 and Deb, Pratap, Agarwal, & Meyarivan, 2002), provides for the elitism mechanism. In addition to the EAs, there are several multi-objective metaheuristic approaches, such as multi-objective ant colony optimization (MOACO) (McMullen, 2001) and MOPSO (Moore & Chapman, 1999), which have also been used to solve multi-objective optimization problems. A review of multi-objective meta-heuristic approaches is presented by Jones, Mirrazavi, and Tamiz (2002).

Since the development of EAs, there has been a growing interest in obtaining the Pareto optimal solutions using different EAs. A comprehensive review of evolutionary-based multi-objective optimization techniques is provided by Fonseca and Fleming (1995), Coello (1999) and Zhou et al. (2011). Particularly, NSGA, developed by Srinivas and Deb (1994), is a popular algorithm, which uses a non-dominated sorting procedure, applies a ranking method that emphasizes those good solutions, and tries to maintain them in the population. Through a sharing method, this algorithm maintains the diversity in the population. The algorithm explores different regions in the Pareto front and is very efficient in obtaining sufficient Pareto optimal sets. Although NSGA is an effective algorithm, it has been generally criticized for its computational complexity, lack of elitism, and for choosing the optimal parameter value for a sharing parameter. A modified version of NSGA, called NSGA-II, developed by Deb et al. (2000) and Deb et al. (2002), utilizes a fast non-dominated sorting genetic algorithm. This method is computationally efficient, non-elitism preventing, and less dependent on a sharing parameter for diversity preservation. Recently, a reference-point based multi-objective NSGA-II algorithm (called NSGA-III) is proposed by Deb and Jain (2014), which is more efficient to solve problems with more than two objectives.

Since the MOPSO algorithm was first presented by Moore and Chapman (1999), several versions of MOPSO have been proposed in the literature. A comprehensive survey of MOPSO-based algorithms is provided by Sierra and Coello (2005). Later, an experimental comparison with the aim of analyzing the search capabilities of

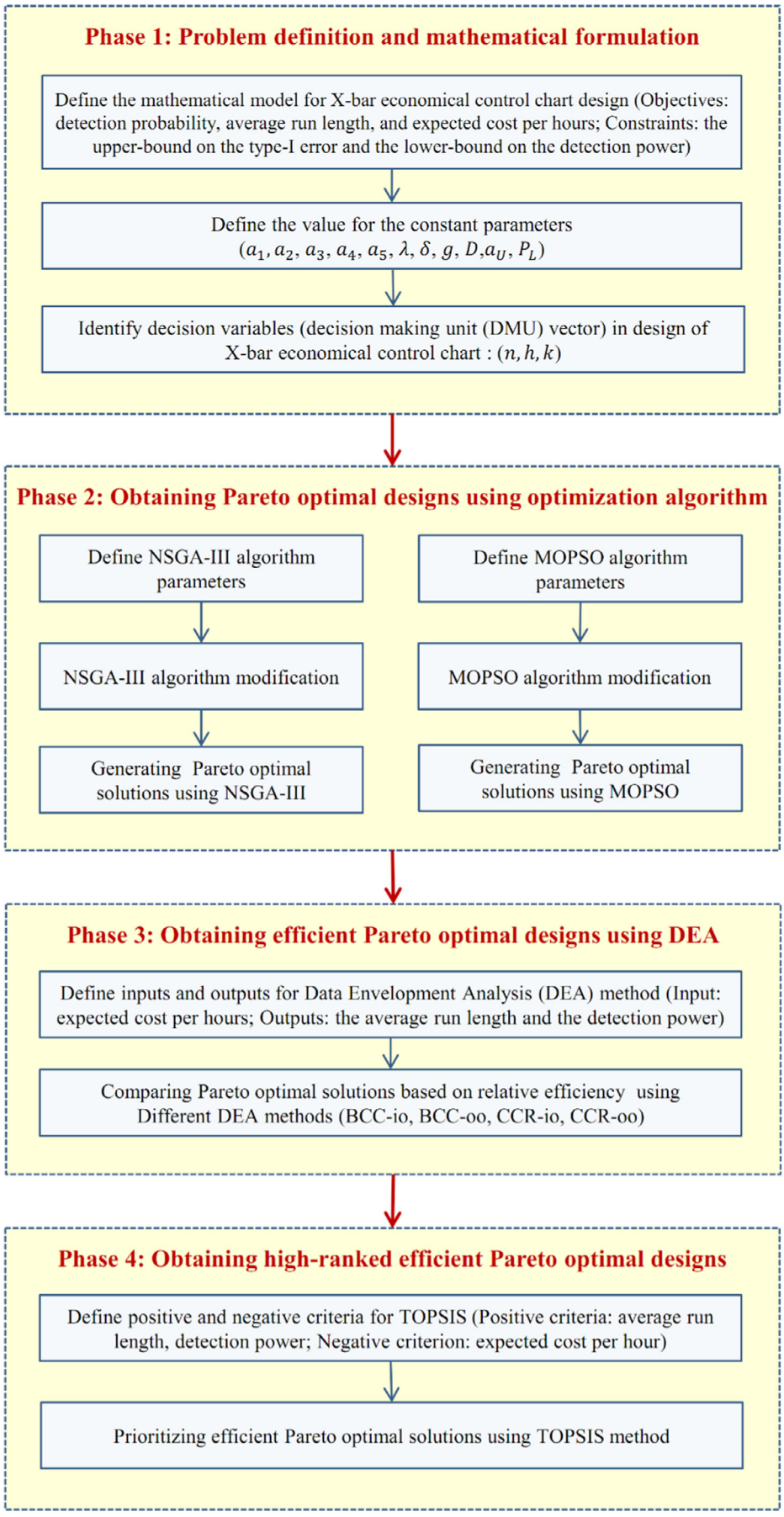


Fig. 2. multi-objective X-bar economical control chart algorithm flowchart.

```
Input:
P_0 (Initial Population),
N_Pop size of population,
t (iteration) = 0,
It_max (Maximum iteration).
While t < It_max do
    Create Offspring Q_t
    Mutation on Q_t
    Set R_t = P_t ∪ Q_t
    Apply non-dominated sorting on R_t and find F_1, F_2, …
    S_t = {}, i = 1;
    While |S_t| ≤ N_Pop do
        S_t = S_t ∪ F_i
        i = i + 1
    End
    IF |S_t| = N_Pop do
        P_{t+1} = S_t; break
    Else
        P_{t+1} = ⋃_{j=1}^{l-1} F_j
        Normalize S_t using min and intercept points of each objective
        Associate each member of S_t to a reference point
        Choose N_Pop − |P_{t+1}| members from F_l by niche-preserving operator
    End
t = t + 1
End
Report P_t
```

Fig. 3. NSGA-III pseudo-code.

six different versions of MOPSO was conducted by Durillo et al. (2009). They proposed a new version of MOPSO, which incorporates a velocity constraint mechanism to handle the difficulties of other MOPSOs when facing some multi-frontal problems. Kennedy, Kennedy, Eberhart, and Shi (2001) have shown that MOPSO performs better in solving various benchmark problems, when compared to some evolutionary multi-objective optimization algorithms including NSGA-II. Particle swarm optimization (PSO) has also been offered as a suitable algorithm for multi-objective optimization mainly because of the high speed of convergence that the algorithm exhibits for single-objective optimization (Kennedy et al., 2001). In this study, two efficient metaheuristic algorithms, NSGA-III (Deb & Jain, 2014) and MOPSO (Coello, Pulido, & Lechuga, 2004) are adopted to solve the control chart deisgn problem. The details of these algorithms are presented in the following subsections.

We begin by defining several basic concepts for NSGA-III and MOPSO algorithms (Coello et al., 2004). A point $\vec{x}^* \in \Omega$ is Pareto Optimal if for every $\vec{x} \in \Omega$ and $I = \{1, 2, \ldots, k\}$ either $\forall_{i \in I}(f_i(\vec{x}) = f_i(\vec{x}^*))$, or there is at least one $i \in I$ such that $f_i(\vec{x}) > f_i(\vec{x}^*)$; where function $f : \Omega \subseteq R^n \to R$, $\Omega \neq \emptyset$, $\vec{x}$ and $\vec{x}^* \in \Omega$, and $\vec{x}^*$ is the global minimum solution, and the set Ω is the feasible region ($\Omega \in S$, where S represents the whole search space.) In other words, $\vec{x}^*$ is Pareto Optimal when there is no feasible vector $\vec{x}$, which would decrease some criterion without causing a simultaneous increase for at least one other criterion. The Pareto Dominance concept is based upon a vector $\vec{u} = (u_1, \ldots, u_k)$ which is said to dominate $\vec{v} = (v_1, \ldots, v_k)$, denoted by $\vec{u} \preccurlyeq \vec{v}$, if and only if u is partially less than v. In other words, $\forall i \in \{1, \ldots, k\}$, $u_i \leq v_i \wedge \exists i \in \{1, \ldots, k\} : u_i < v_i$. For a given multi-objective problem ($\vec{f}(x)$), the Pareto optimal set P^* is defined as: $P^* := \{x \in \Omega | \neg \exists x' \in \Omega \vec{f}(x')^\circ \ \vec{f}(x)$; and the Pareto front (PF^*) is defined as $PF^* := \{\vec{u} = \vec{f} = (f_1(x), \ldots, f_k(x)) | x \in P^*\}$.

3.1. NSGA-III

As indicated in the previous sections, a new approach, called NSGA III, is incorporated in the selection mechansim of NSGA-II. The idea is to use reference points which could be a set of predefined points, or one that are generated systematically. The pseudo-code of NSGA-III is shown in Fig. 3 (Deb & Jain, 2014).

The algorithm starts with N_{Pop} where Po denotes the initial population . As presented in Section 2, n, h, and k are the parameters of the existing problem. Each individual (solution) in the population P_0, as well as in the populations of the next generations, is represented by $s_i = (n_i,\ h_i,\ k_i)$ for $i = 1, .., N_{Pop}$. Note that in this study, individuals of the initial population are randomly generated, such that $n_i \in \mathbb{N}^+$ and $h_i,\ k_i \in \mathbb{R}^+$, for $i = 1, .., N_{Pop}$, as follows:

$$n_i = [n_{min} + rand.(n_{max} - n_{min})] \tag{11}$$

$$h_i = h_{min} + rand.(h_{max} - h_{min}) \tag{12}$$

$$k_i = k_{min} + rand.(k_{max} - k_{min}) \tag{13}$$

where x_{min} and x_{max} represent lower and upper bounds for the variable x, respectively, rand is a uniform random number between 0 and 1, and $[x]$ represents the smallest integer greater than the real number x.

The next step of NSGA-III is to generate the offspring Q_t. In this study, an arithmetic crossover operator proposed by Michalewicz (1996) is used where two individuals s_i and s_r from the current population are randomly selected to generate two offspring q_i and q_r, as follows:

$$q_i = (\beta)s_i + (1 - \beta)s_r \tag{14}$$

$$q_r = (1 - \beta)\ s_i + (\beta)\ s_r \tag{15}$$

where β is a uniform random number in the range [0,1].The generated offspring q_i and q_r are added to Q_t. The next step is to use a mutation operator, which is applied to the new offspring. In this study, the mutation operator is assumed to be Gaussian, which results in more mutation at the beginning of the algorithm compared with the end of the algorithm.

Each solution (individual) should satisfy the constraints of problems represented by Eqs. (9)–(10). In the proposed NSGA-III, the suggested method by Jain and Deb (2014) is used in conjunction with

the constraints, i.e., the constraints are normalized as represented by Eqs. (16)–(18), then for each individual s_i the constraint violation value ($CV(s_i)$) is calculated by Eq. (15), where x is $-x$ if $x < 0$, and 0 otherwise.

$$g_1(s_i) = \frac{p(s_i)}{p_L} - 1 \geq 0, \quad \forall s_i \tag{16}$$

$$g_2(s_i) = -\alpha(s_i)/\alpha_U - 1 \geq 0, \ \forall s_i \tag{17}$$

$$CV(s_i) = g_1(s_i) + g_2(s_i) \tag{18}$$

In the next step, the parent population P_t and the offspring Q_t are combined as R_t with a size of $2 * N_{Pop}$. Then, the fast non-dominated sorting based on Pareto dominance is applied to R_t to classify it into different non-dominance levels, i.e., F_1, F_2, and so on. Depending on the feasibility and infeasibility of the R_t members, there are three possible scenarios to classify individuals into different non-dominance levels, as follows:

- *All members of R_t are infeasible*: In this case, individuals (solutions) with the smallest constraint violations, $CV(s_i)$, will be in the first non-dominated level, F_1, and individuals with the second smallest violations will be in the second non-dominated level, F_2, and so on.
- *All members of R_t are feasible*: In this case, the regular non-dominating sorting procedure, presented by Deb et al. (2002), is applied.
- *Some of R_t members are infeasible (and the rest of them are feasible)*: In this case, the feasible solutions based on the standard non-dominating sorting procedure will be assigned to the first levels, and then the infeasible solutions will be assigned to the higher levels similar to the case when all members of R_t are infeasible.

After finding different levels of non-dominance levels $F_1, F_2, \ldots$, the next step in the NSGA-III algorithm is to generate the next generation P_{t+1} (based on $F_1, F_2, \ldots$). Starting from F_1, the individuals in the higher non-dominance levels are added to S_t until its size reaches N_{Pop} or exceeds N_{Pop} for the first time, assuming the non-dominance level l. Individuals (solutions) in the non-dominance levels greater than l are simply discarded, and $S_t \backslash F_l$ are selected as the next generation P_{t+1}. If the size of the next generation P_{t+1} is N_{Pop}, the algorithm then repeats the previous step in the next iteration by generating new offspring (if the stopping condition of the algorithm is not met), otherwise, the other $N_{Pop} - |P_{t+1}|$ individuals (solutions) are chosen from F_l based on reference points. Since the objectives may be based upon different scales, they are normalized and reference points are generated in the normalized space. Then, each solution (individual) is assigned to a reference point. Each individual in $S_t \backslash F_l$ and F_l is then assigned to the closest reference point and the remaining $N_{Pop} - |P_{t+1}|$ individuals in F_l are chosen so that their associated reference point does not have any associated individual in $S_t \backslash F_l$. In the following paragraphs, the details of objectives normalization and selection of the remaining $N_{Pop} - |P_{t+1}|$ individuals in F_l are discussed in more detail.

Normalizing the objectives: Each objective f_j is normalized as shown in Eq. (19):

$$f_j^n = \frac{f_j(.) - z_j^{min}}{a_j - z_j^{min}}, \quad j = 1, 2, 3 \tag{19}$$

where $f_j(.) - z_j^{min}$ is the translated objective, $z_j^{min} = \min_{s_i \in S_t} f_j(s_i)$, a_j is intercept point of objective f_j, and f_j^n denotes the normalized objective f_j.

In order to find the intercept points, a_j, the extreme point z_j^{max} in each translated objective axis needs to be identified based on Eq. (20), where s^* is determined based on Eqs. (21) and (22). Note that w is $w = (\epsilon, \epsilon, \epsilon)$ where $\epsilon = 10^{-6}$ and $w_j = 1$.

$$z_j^{max} = f_j(s^*) - z_j^{min} \tag{20}$$

$$s^* = \ argmin_{s_i \in S_t} ASF(s_i, \boldsymbol{w}) \tag{21}$$

$$ASF(s_i, \boldsymbol{w}) = \max_{j=1,2,3} \left(f_j(s_i) - z_j^{min}\right)/w_j \tag{22}$$

After finding z_j^{max}, Eq. (23) is used to find a_j (Yuan, Xu, Wang, & Yao, 2015).

$$\begin{pmatrix} \left(a_1 - z_1^{min}\right)^{-1} \\ \left(a_2 - z_2^{min}\right)^{-1} \\ \left(a_3 - z_3^{min}\right)^{-1} \end{pmatrix} = E^{-1}u \tag{23}$$

where $E = (z_1^{max} - Z, z_2^{max} - Z, z_3^{max} - Z)^T$, and $Z = (z_1^{min}, z_2^{min}, z_3^{min})$, and $u = (1, 1, 1)^T$.

If the rank of the matrix E is less than 3, E^{-1} is undefined, and therefore, the resulting extreme points cannot construct the hyper-plane (Yuan et al., 2015). If the matrix E is not a full rank, simply set $a_j = \max_{s_i \in S_t} f_j(s_i)$.

Generating reference points: The novelty of the NSGA-III algorithm is the use of reference points to select individuals and preserve the diversity of the population. The reference points are either provided by experts, the knowledge about the problem at hand, or generated systematically (e.g., such as Das and Dennis's (1998) method). Each member of S_t is associated with the reference point which has the closest Euclidean distance from it. The goal is to assign higher priority to those reference points in F_l that are not well represented in $S_t \backslash F_l$ to be in the next generation P_{t+1}.

In this study, the method developed by Das and Dennis (1998) is used to generate the reference points $\Re$. According to this method, the reference points are placed on the normalized hyper-plane, where the number of reference points H depends on the number of objectives (m), which is assumed to be 3 in this study ($m = 3$), as well as the given division $g \in \mathbb{N}^+$, which determines the distance of the reference points from each other on the same surface. The total number of reference points is determined as shown in Eq. (24).

$$H = \binom{m + g - 1}{m - 1} \tag{24}$$

The two-layered reference point is used in this paper as suggested by Deb and Jain (2014) and Jain and Deb (2014). A hypothetical example of two-layered reference points, where both layers have the same number of points, is shown in Fig. 4. Suppose g_1 and g_2 represent the divisions of outer and inner layers, respectively, then the total number of reference points is as shown in Eq. (25).

$$H = \binom{m + g_1 - 1}{m - 1} + \binom{m + g_2 - 1}{m - 1} \tag{25}$$

Niche-Preserving *Operation*: After generating the reference points, the distance of each individual from the reference line, the line that connects the origin of the space of the normalized objectives to the reference point, are calculated. Then, each individual is assigned to the closest reference point. Fig. 4 illustrates the reference lines of the reference points.

In the end, some reference points may have one or more associated members and some may not have any associated members. Those members of $S_t \backslash F_l$, associated with the reference point $j \in \Re$, are counted and represented by the niche count ρ_j. In order to obtain a diverse population, the following procedure is used:

First, the reference points with the minimum ρ_j are identified; in case there is more than one reference point, the tie is broken randomly. $\rho_j = 0$ indicates that no member of $S_t \backslash F_l$ is associated with the reference point j. In this case, there are two possible situations: (I) there are members associated with this reference point in F_l, therefore, assign the member with the minimum perpendicular distance to

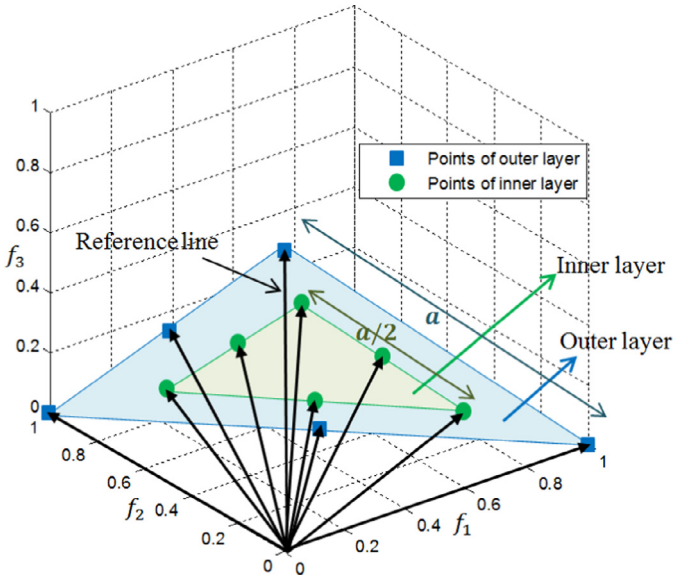


Fig. 4. A hypothetical example of two-layered reference points ($g_1 = g_2 = 2$).

the next generation, P_{t+1}, and set $\rho_j = \rho_j + 1$; (II) there is no member associated to this reference point in F_l, therefore, simply go to another reference point with the minimum ρ_j. If $\rho_j \geq 1$, the reference point j has more than one associated member, and in this case, if there is a member in F_l associated with the reference point j, add the individual with the minimum perpendicular distance to the next generation, P_{t+1}, and set $\rho_j = \rho_j + 1$. The above procedure is repeated until the population size reaches N_{Pop}.

3.2. Multi-objective particle swarm optimization (MOPSO)

Particle swarm optimization (PSO) is inspired by the social behavior of birds within a flock. A particle represents each potential solution of the problem and a swarm represents the population of solutions. In PSO, each particle (solution) searches the solution space based on its current position and velocity direction, where the search is affected by the history of the particle and other individuals. The efficiency and fast convergence of the PSO in solving single objective problems has been extended to solve multi-objective problems (Kennedy et al., 2001). Some extended versions of the MOPSO algorithm are presented and compared by Durillo et al. (2009).

In this study, the MOPSO developed by Coello et al. (2004) is modified to solve multi-objective X-bar economical control chart design problems. The pseudo-code of the general MOPSO is presented in Fig. 5 (Coello et al., 2004).

This algorithm starts with N_{Pop} randomly generated particles (solutions) where each solution has its current position and velocity. In this paper, $s_i^t = \{n_i^t, h_i^t, k_i^t\}$ and $v_i^t = \{v_{i,1}^t, v_{i,2}^t, v_{i,3}^t\}$ represent position and velocity of particle $i = 1, \ldots, N_{Pop}$ at iteration $t = 1, \ldots, It_{max}$, respectively, where the initial position and velocity of particles, s_i^0 and v_i^0, are generated by Eqs. (11)–(13). Then, each individual is evaluated based on its fitness (goodness). Non-dominated solutions (particles) are stored in an external repository (or archive) (called REP hereinafter) (Coello et al., 2004). The REP has two control components: an archive controller and a grid. The former dictates if a solution should be added to the REP or not. At each iteration of the algorithm, the non-dominated solutions are compared one by one to the solutions in the REP. If the new solution is dominated by any member of the REP, the solution then will be discarded; otherwise, the new solution will be added to the REP. After adding the new solution, if there are any solutions in the REP dominated by the new solution, those solutions will be discarded. Since the capacity of the REP is limited, when it reaches its limit, the decision about adding a new solution is made using the adaptive grid (Coello et al., 2004). The ultimate goal of using the adaptive grid is having well-distributed Pareto fronts. The space of the objectives is divided into regions that would be changed depending on the solutions in the REP, i.e., when a new solution is found outside of the current grid, the grid will be updated and the individual within it will be relocated. Without losing generality, one can consider the grid as connected hypercubes and each hypercube can have some individuals in it while some of them may be empty.

At each iteration, each particle moves toward the best position it has encountered so far and the best position found by the swarm until that point of the procedure. In this paper, $pBest_i^t$ represents the best position found by particle i up to iteration t, and $gBest^t$ indicates the best position found by the swarm up to iteration t. The next step of the algorithm is finding (updating) $pBest_i^t$ of each particle and $gBest^t$ of the swarm, which are represented by Eqs. (26) and (27)

```
1.   Input:
     P_0 (Initialize Swarm)
     N_Pop size of population,
     t (Generation) = 0
     It_max (Maximum iteration)
2.   Record non-dominated particles in REP
3    Generate the grid (hypercubes)
4.   Update pBest_i^t
5.   Update gBest^t
6.   While t < It_max do
7.        For each particle i do
8.             Update velocity v_i^t
9.             Update new position s_i^t
10.            Update pBest_i^t
11.       End For
12.       Update gBest^t
13.       Update REP
14.       t = t + 1
15.  End While
```

Fig. 5. MOPSO pseudo code.

(Coello et al., 2004):

$$pBest_i^t = \left\{xBest_{i,1}^t, xBest_{i,2}^t, xBest_{i,3}^t\right\} \tag{26}$$

$$gBest^t = \left\{gBest_1^t, gBest_2^t, gBest_3^t\right\} \tag{27}$$

where $gBest^t$ is a randomly chosen individual from the REP within the hypercube, which has the fewest individuals. To select such a hypercube, the fitness of individuals located in the hypercubes with more than one individual is divided by a positive number (in this study 10). Then an individual is randomly chosen from the selected hypercube using a roulette-wheel selection mechansim.

Each particle has its own velocity which moves toward $pBest_i^t$ and $gBest^t$. The velocity of the particle i at each iteration t changes based on its velocity at the pervious iteration and $pBest_i^t$ and $gBest^t$ as represented by Eq. (28) (Coello et al., 2004).

$$v_i^t = w_t * v_i^{t-1} + C_1 * r_1 * \left(pBest_i^t - s_i^{t-1}\right) + C_2 * r_2 * \left(gBest^t - s_i^{t-1}\right) \tag{28}$$

where r_1 and r_2 are uniformly distributed random numbers between 0 and 1, and C_1 and C_2 are acceleration constants that control the effect of $pBest_i^t$ and $gBest^t$, respectively, on the speed of the particle. Also, w_t is an inertia weight of the particle i and controls the trade-off between the global and local history. w_t has a large impact on the performance of MOPSO. Thus, the gradually decreasing linear inertia weight as suggested by Shi and Eberhart (1999) and Naka, Genji, Yura, and Fukuyama (2001) is applied as follows:

$$w_t = w_{max} - \frac{w_{max} - w_{min}}{It_{max}} t \tag{29}$$

where t and Itmax represent the current iteration of the algorithm and maximum iteration of the algorithm, respectively, and w_{min} and w_{max} are the lower and upper bounds of w_t set to $w_{min} = 0.4$ and $w_{max} = 0.9$ as shown in Shi and Eberhart (1999) and Naka et al., 2001.

After finding the velocity of each particle, the position of particle i is updated as shown in Eq. (30).

$$s_i^t = s_i^{t-1} + v_i^t \tag{30}$$

Note that it is possible for s_i^t to fall outside of the feasible boundary. In such a case, there are two possible options: (I) set the boundary value of the variable as the current position of the particle, and (II) multiply v_i^t by (-1) to move the particle in the opposite direction. Recall that $n_i \in \mathbb{N}^+$, and if n_i^t is not integer, then simply generate two solutions, one is based on rounding up the value of n_i^t, and the other one is based on rounding down the value of n_i^t. Then, these two solutions are evaluated, and the dominant solution is kept, and the dominated one is discarded.

In order to handle the constraints of the multi-objective problem presented in this paper, a simple strategy has been used in the MOPSO algorithm. When two solutions are compared, if both are feasible, use the regular dominancy definition. If one is feasible and the other one is infeasible, the feasible solution dominates the other. If both of them are infeasible, the one that has the less amount of constraint violation dominates the other.

If the stopping condition of the algorithm is not been met, the algorithm starts another iteration by updating the position and velocity of the particles; otherwise, the solutions in the REP will be repeated.

4. MCDM tools to reduce and rank the optimal solutions

Even though the results of multi-objective optimization algorithms are informative, the number of solutions may still be prohibitive for a decision-maker to make choices. At this point, selecting representative solutions from all the solutions obtained from the algorithms can be solved as a multi-objective optimization problem, which can significantly reduce the size of the solutions from the Pareto optimal solutions. The DEA method is a special approach that can achieve this goal by eliminating the inefficient Pareto optimal solutions. Efficient Pareto optimal solutions can be presented to the decision-makers, and those designs (or solutions), which are not satisfactory in terms of the relative efficiency can be easily ommited. Since all efficient solutions obtained by DEA have the same level of efficiency, there is still a need to find the best solution from the efficient optimal solutions. TOPSIS can be used to rank the optimal efficient solutions and accomplish this goal. The DEA and TOPSIS procedures are presented in the following subsections.

4.1. Pareto optimal solution reduction using DEA

DEA, introduced by Charnes, Cooper, and Rhodes (1978), is a method for measuring the relative performance of the DMUs. This method is based on linear programming methods and compares the relative efficiencies of the DMUs, which use multiple inputs (i.e., cost type criteria) to produce multiple outputs (i.e., benefit type criteria) (Cook, Tone, & Zhu, 2014). Each alternative solution is considered as a DMU in the DEA method, and all the DMUs are generally assumed to be homogeneously comparable, so that the resulting relative efficiencies can be compared on the same scale. The relative efficiency incorporating multiple inputs and outputs can be defined (Charnes et al., 1978) in order to compare their efficiencies.

In DEA, a ratio of a weighted sum of outputs to a weighted sum of inputs is calculated as a measure of efficiency for each DMU. Consider a set of n DMUs, with each DMU j, $(j = 1, \ldots, n)$ using m inputs x_{ij} $(i = 1, \ldots, m)$ and generating s outputs $y_{rj} (r = 1, \ldots, s)$. If the weights (price or multipliers) $\bar{u}_r$ and $\bar{v}_i$ associated with output r and input i, respectively, are known, the efficiency $(\bar{e}_j)$ of DMU_j, as the ratio of weighted outputs to weighted inputs, is equal to $\sum_r u_r y_{rj} / \sum_i v_i x_{ij}$. Charnes et al. (1978) proposed their DEA model (known as CCR) which is the constant returns to scale model in the absence of known multipliers. Their model measures the technical efficiency of DMU_0 by solving the following fractional programming problem, as represented by Eq. (31). In this model, known as the original CCR input-oriented model, ε is a non-Archimedean element smaller than any positive real number.

$$\begin{aligned} & e_0 = \max \sum_r u_r y_{r0} \Big/ \sum_i v_i x_{i0} \\ & \text{s.t.} \sum_r u_r y_{rj} - \sum_i v_i x_{ij} \le 0, \ \ all\ j \\ & u_r, v_i \ge \varepsilon, \ \ \forall\, r, i. \end{aligned} \tag{31}$$

Since this model considers the ratio of outputs to inputs, it is referred to as the input-oriented model. The output oriented model is the inverted form of this ratio with a minimization objective.

By making the change of variables $\mu_r = t u_r$, and $v_i = t v_i$, where $t = (\sum_i v_i x_{i0})^{-1}$, the aforementioned fractional programming problem can be converted to a linear programming model, known as the envelopment or primal problem, as represented by Eq. (32):

$$\begin{aligned} & e_0 = \max \sum_r \mu_r y_{r0} \\ & \text{s.t.} \quad \sum_i v_i x_{i0} = 1 \\ & \sum_r \mu_r y_{rj} - \sum_i v_i x_{ij} \ \le 0, \ \forall j \\ & \mu_r, \ v_i \ \ge \ \varepsilon, \ \forall\, r, i. \end{aligned} \tag{32}$$

The duality of the previous problem is a linear programming problem, known as the multiplier or dual problem, as represented by

Eq. (33), which provides detailed information for the relative efficiency measure, where S_i^- and S_r^+ are slack variables (Cook & Seiford, 2009).

$$\begin{aligned}
&\min \theta_0 - \varepsilon\left(\sum_r S_r^+ + \sum_i S_i^-\right)\\
&\text{s.t.} \quad \sum_j \lambda_j x_{ij} + S_i^- = \theta_0 x_{i0},\ i = 1,\ldots,m\\
&\sum_j \lambda_j y_{rj} + S_r^+ = y_{r0},\ r = 1,\ldots,s\\
&\lambda_j, S_i^-, S_r^+ \geq 0,\ \forall i, j, r\\
&\theta_0 \textit{ unconstrained}
\end{aligned} \tag{33}$$

The other DEA model (the BCC model), which was introduced by Banker (1984), is the extension of the CCR model and is basically the variable returns to scale model. The BCC model differs from the CCR model, by way of an additional variable (u_0).

The linear programming of the BBC model is presented in Eq. (34) and the dual of this BCC model is represented by Eq. (35).

$$\begin{aligned}
&e_0^* = \max \sum_r \mu_r y_{r0} - \mu_0\\
&\text{s.t.} \quad \sum_i v_i x_{i0} = 1\\
&\sum_r \mu_r y_{rj} - \mu_0 - \sum_i v_i x_{ij} \leq 0,\ j = 1,\ldots,n\\
&\mu_r, v_i \geq \varepsilon,\ \forall r, i. \qquad \mu_0 \textit{ unrestricted in sign.}
\end{aligned} \tag{34}$$

$$\begin{aligned}
&\min \theta_0 - \varepsilon\left(\sum_i s_i^- + \sum_r s_r^+\right)\\
&\text{s.t.} \quad \sum_j \lambda_j x_{ij} + S_i^- = \theta_0 x_{i0},\ i = 1,\ldots,m\\
&\sum_j \lambda_j y_{r0} - S_r^+ = y_{r0},\ r = 1,\ldots,s\\
&\sum_j \lambda_j = 1\\
&\lambda_j, S_i^-, S_r^+ \geq 0\ \forall i, r, j \qquad \theta_0 \textit{ unrestricted in sign.}
\end{aligned} \tag{35}$$

The dual of BCC differs from the dual of CCR because it has additional convexity constraints on λ_j ($\sum_j \lambda_j = 1$). In both the CCR and BCC models, the performance of DMU_0 is fully (100%) efficient if and only if both (I) $\theta_0^* = 1$ and (II) all slacks $S_i^{-*} = S_r^{+*} = 0$, and weakly efficient if and only if both (I) $\theta_0^* = 1$ and (II) $S_i^{-*} \neq 0$ and/or $S_r^{+*} \neq 0$ for some i and r for some alternative optima. Clearly, any CCR-efficient DMU is also BCC-efficient, but BCC-efficient solutions may not be CCR-efficient. Thus, we would expect more efficient solutions from the BCC model and fewer efficient solutions from the CCR model. The CCR model is referred to as giving a radial projection. Particularly, each input is reduced by the same proportionality factor θ. The BCC model provides more of a flexible projection by providing decreasing, increasing and constant returns to scale the frontier (Cook & Seiford, 2009).

4.2. TOPSIS to prioritize the optimal solutions

TOPSIS was originally introduced by Hwang and Masud (1979). The basic mechanism of this approach is to calculate the distance from each alternative to a positive ideal solution (PIS) and a negative ideal solution (NIS) that are defined in n-dimensional space, where n is the number of criteria in the decision problem. The best alternative should have the smallest vector distance from the PIS and the greatest from the NIS. There are several different methods for determining the PIS and NIS. Here, the PIS can be defined as the best performing attribute for each criterion amongst the alternatives. Conversely, NIS can be constructed from the poorest performing attributes of each criterion amongst the alternatives. This process is presented below in five steps (Salmon, Mobin, & Roshani, 2015; Mobin et al., 2015). In Step 1, alternative data is normalized via vector normalization (Eq. (36)), where x_{ij} is the appraisal matrix R of alternative i under appraisal criterion j and r_{ij} is the normalized appraisal matrix R of alternative i under appraisal criterion j.

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}},\ i = 1,2,\ldots,m;\ j = 1,2,\ldots,n \tag{36}$$

In Step 2, these normalized values are then weighted via Eq. (37), where v_{ij} is the weighted normalized appraisal matrix R of alternative i under appraisal criterion j.

$$v_{ij} = r_{ij} * w_j,\ i = 1,2,\ldots,m;\ j = 1,2,\ldots,n \tag{37}$$

In Step 3, Eqs. (38) and (39) are used to determine PIS (A^+) and NIS (A^-), where A_j^* is the PIS for the j_{th} criteria and A_j^- is the NIS for the j_{th} criteria.

$$\begin{aligned}
A_j^* &= \{(Max_i v_{ij} |\ j \in J),\ (Min_i v_{ij} |\ j \in J') \mid i = 1,2,\ldots,n\}\\
&= \left\{v_1^*,\ v_2^*,\ldots,v_j^*,\ldots,v_n^*\right\}
\end{aligned} \tag{38}$$

$$\begin{aligned}
A_j^- &= \{(Min_i v_{ij} |\ j \in J),\ (Max_i v_{ij} |\ j \in J') \mid i = 1,2,\ldots,n\}\\
&= \left\{v_1^-,\ v_2^-,\ldots,v_j^-,\ldots,v_n^-\right\}
\end{aligned} \tag{39}$$

In Step 4, the n-criteria evaluation distance can measure the separation from the PIS and NIS for each alternative.

$$S_i^* = \sqrt{\sum_{j=1}^n \left(v_{ij} - v_j^*\right)^2},\ i = 1,\ 2,\ldots,\ n \tag{40}$$

$$S_i^- = \sqrt{\sum_{j=1}^n \left(v_{ij} - v_j^-\right)^2},\ i = 1,\ 2,\ldots,n \tag{41}$$

Finally, in Step 5, the relative closeness to the ideal solution (C_i^*) is calculated by Eq. (42). The relative closeness of the i_{th} alternative, with respect to the PIS, is defined as C_i^*. If the value of C_i^* is closer to 1, the alternative i will be closer to the PIS. The alternative with the highest C_i^* will get the first rank and so on.

$$C_i^* = \frac{S_i^-}{S_i^* + S_i^-},\ 0 < C_i^* < 1,\ i = 1,2,\ldots,n \tag{42}$$

5. Case study

In this section, the proposed integrated optimization method is applied to the industrial case, borrowed from Pearn and Chen (1997) and also reinvestigated by Chen and Liao (2004) and Mobin et al. (2015). The case study is about the process of producing electrolytic capacitors, where the target value of capacitance, for a particular model is set to 300 (in μF). The process shifts occur at random with a frequency of about 1 every 4 hours of operation ($\lambda = 0.25$). The fixed cost of sampling is estimated to be \$1.00 ($a_1 = 1$) and the variable cost is assumed to be \$0.1 per capacitor ($a_2 = 0.1$). The average time of sampling, measuring and recording the capacitance is estimated to be 0.01 h ($g = 0.01$).When the process goes out of control, the magnitude of the shift is approximately estimated to be one standard deviation ($\delta = 1.0$). The average time to search the assignable cause is 2 h ($D = 2$). The cost to search the assignable cause and also the measurable portion of the cost to investigate the false alarm are both \$50 ($a_3 = a_4 = 50$). The penalty cost associated with production in the OOC state is considered to be approximately

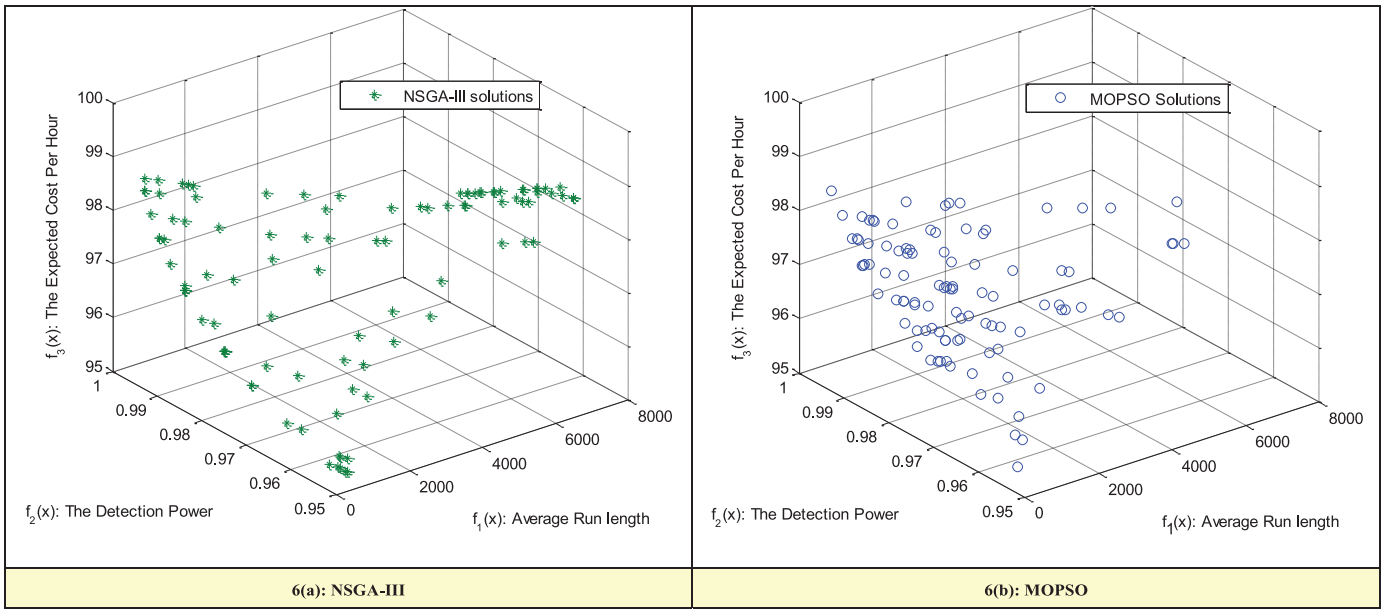


Fig. 6. Pareto optimal frontier for the multi-objective X-bar economical control chart problem.

\$200 per hour ($a_5 = 200$). Based on quality control experts' suggestions, the upper bound on the type-I error and the lower bound of the detection power are assumed to be 0.005 and 0.95 respectively ($\alpha_U = 0.005,\ p_L = 0.95$). The multi-objective X-bar economical control chart design formulation for this case study is presented in Eq. (43).

$$
\begin{aligned}
&f_1: \max_s ARL_0(s) = \max_s \frac{1}{2\int_{-\infty}^{-k}\Phi(z)dz}\\
&f_2: \max_s p(s) = \int_{-\infty}^{-k-\delta\sqrt{n}}\Phi(z)dz + \int_{k-\delta\sqrt{n}}^{\infty}\Phi(z)dz\\
&f_3: \min_s E_{HC}(s) = \frac{E_{CC}(s)}{E_{CT}(s)} = \frac{\frac{(a_1+a_2n)E_{CT}(s)}{h} + a_3 + \frac{a_4\alpha\exp(-\lambda h)}{1-\exp(-h\lambda)} + a_5\left(\frac{h}{p(s)} - \tau + gn + D\right)}{\frac{1}{\lambda} + \left(\frac{h}{p(s)} - \tau\right) + gn + D}\\
&\text{s.t.}\\
&\alpha(s) \le \alpha_U,\\
&\forall s = (n,h,k), \quad \{n \in \mathbb{Z},\ 20 \le n \le 30\},\\
&\qquad \{h \in \mathbb{R},\ 0.4 \le h \le 0.5\} \text{ and } \{k \in \mathbb{R},\ 2.9 \le k \le 3.8\}
\end{aligned}
\tag{43}
$$

where $\Phi(z)$ can be obtained from the prepared table of the standard normal cumulative distribution function or from the following formulation: $\Phi(z) = \int_{-\infty}^{z}\frac{1}{\sqrt{2\pi}}e^{-u^2/2}du$. The value of τ can be calculated by $\tau = \frac{1-(1+\lambda h)e^{-\lambda h}}{\lambda(-e^{-\lambda h})}$, and $\alpha(s) = \frac{1}{ARL_0(s)} = 2\int_{-\infty}^{-k}\Phi(z)dz$.

5.1. Pareto frontiers obtained by NSGA-III and MOPSO

In this subsection, the proposed NSGA-III and MOPSO algorithms, presented in Section 3, are used to solve the presented numerical example based on the foregoing mathematical model (Eq. (43)). There are three decision variables in this multi-objective X-bar economical control chart design, represented by the vector $s = (n, h, k)$. The range of the decision variables are considered to be same as in Chen and Liao (2004) and Mobin et al. (2015). The range of sample size (n) is confined from 10 to 20. Large sample sizes are not considered because of high inspection expenditures. Similarly, the range of the sampling time interval (h) is set from 0.4 to 0.5 hours, while the range for the control limit width is from 2.9 to 3.8 of the standard deviation σ.

We modify the NSGA-III and MOPSO algorithms to initiate both continuous and discrete decision varibles since the multi- objective design of the control chart problem is a mixed-integer problem, that is, n is a discrete decision variable and h and k are continuous decision variables. In addition, a variable fixing rule is applied in all iterations of the algorithms if the value of n is not an integer. Finally, the two algorithms are modified to consider three objectives and two constraints in the multi-objective control chart design problem presented in Eq. (43). When evaluating the solutions, the NSGA-III uses the concept of the penalty function while the MOPSO uses a simple strategy where a feasible solution dominates an infeasible solution.

In summary, we take advantage of both the NSGA-III and the MOPSO algorithms in order to obtain an optimal economical X-bar control chart design since they use different search stratgies for exploring the feasible regions and utilize different methods to handle constraints and selection mechanisms.

Before running the proposed NSGA-III algorithm, its parameters need to be specified to control its convergence rate and running time. Based on the initial experiments, the parameters of the algorithms are set as follows: the probability of mutation $p_m = 1/55$, the probability of crossover $p_c = 1$, and $g_1 = 7$ and $g_2 = 4$.

For both the NSGA-III and MOPSO algorithms, 100 solutions are generated (the number of population for each algorithm $N_{Pop} = 100$) and the maximum number of iterations It_{max} is assumed to be 60. Note that the combination of g_1 and g_2, according to Eq. 25, creates 102 reference points, and since the population size is 100, two of the (102) reference points at each iteration are randomly dropped. Multiple runs of the NSGA-III and the MOPSO algorithms for the MODM X-bar economical control chart design formulation generate very stable Pareto frontiers for each algorithm as shown in Fig. 6(a) and (b), respectively.

For comparison purposes, the Pareto frontiers obtained by NSGA-III and MOPSO are combined and plotted in Fig. 7. Both Pareto frontiers fall approximately in the same range, although the NSGA-III frontiers were close to the border area of the feasible solutions and generated more solutions at the edge of the Pareto frontier.

Table 2 presents a summary of the results obtained from the two optimization algorithms. As can be observed, both algorithms generate solutions such that the values of the decision variables and objective functions fall approximately in the same ranges.

5.2. Comparing the performance of the two algorithms

In this section, the performance of the proposed algorithms are compared in terms of the convergence rate and the diversity of solutions based on the available metrics in the literature, as follows.

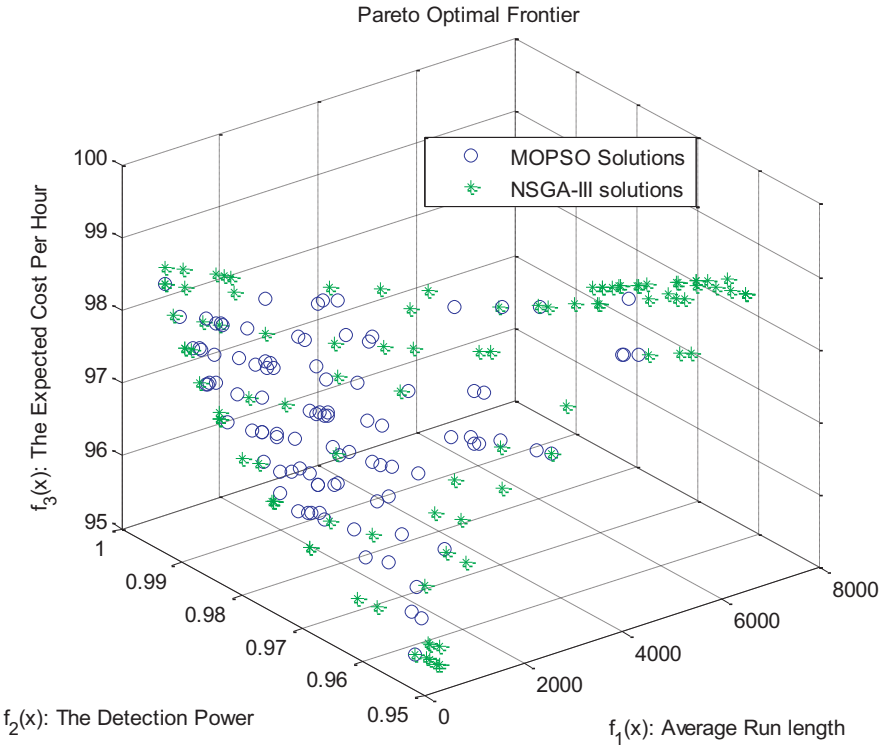


Fig. 7. Combined NSGA-III and MOPSO Pareto frontiers.

Table 2
Summary solutions.

Parameter	NSGA-III results			MOPSO results		
	Minimum	Maximum	Average	Minimum	Maximum	Average
n	21	30	27.56	21	30	26.8
h	0.4	0.48639	0.438813	0.401441	0.499747	0.451195
k	2.9	3.8	3.343374	2.9	3.733528	3.215011
$ARL_0(s)$	267.9797	6911.037	2523.49	267.9797	5296.156	1090.571
$p(s)$	0.950003	0.99502	0.967355	0.950092	0.995019	0.972723
$E_{HC}(s)$	95.32335	99.07514	97.88204	95.33462	98.85651	97.52637

***Metric 1*:** Pareto dominance indicator (Goh & Tan, 2009): Let S_1 and S_2 be the solutions obtained from NSGA-III and MOPSO, respectively; and $B = \{b_i | \forall b_i, \neg\exists a_i \in (S_1 \cup S_2) \prec b_i\}$ where $a_i \prec b_i$ means that a_i dominates b_i. This metric measures the ratio of the non-dominated solutions contributed by S_j for $j = 1, 2$ to all the non-dominated solutions (Eq. (44)).

$$NR(S_j) = \frac{|S_j \cap B|}{|B|}, \quad j = 1, 2 \tag{44}$$

Note that $0 \le NR(S_j) \le 1$ and $NR(S_j) = 0$ means that all the solutions of S_j are dominated by the other set of solutions and $NR(S_j) = 1$ indicates that all the solutions of S_j are in the non-dominated frontier.

***Metric 2*:** Distribution diversity metric (Deb et al., 2000): This metric is defined in Eq. (45).

$$\Delta(S_j) = \sum_{i=1}^{|S_j|-1} \frac{(d_i - \bar{d})}{|S_j| - 1} j = 1, 2 \tag{45}$$

where d_i is the Euclidean distance between consecutive solutions, and $\bar{d}$ is the average of d_i. Solutions are sorted in lexigraphical order. When the distance of all the solutions are equal, then $\overline{d3} = d_i$, implying that $\Delta(S_1) = 0$ and that S_1 has a perfect distribution.

***Metric 3*:** Spacing (Schott, 1995): This metric is an extension of the previous metric.

$$C(S_j) = \sqrt{\sum_{i=1}^{|S_j|} (d_i - \bar{d})^2 / |S_j|}, \quad j = 1, 2 \tag{46}$$

Metric *4*: Convergence of two sets (Zitzler & Thiele, 1999): This measure considers the overlaps between solutions of the algorithms as presented in (47). Note that $0 \le C(S_1, S_2) \le 1$ where $C(S_1, S_2) = 1$ indicates that all the solutions in S_2 are dominated by the solutions in S_1; and $C(S_1, S_2) = 0$ means that none of the solutions in S_2 are dominated by one of the solutions in S_1.

$$C(S_1, S_2) = \frac{|\{s_i \in S_2 | \exists s_r \in S_1 : s_i \prec s_r\}|}{|S_2|} \tag{47}$$

The summary of the calculated metrics for both algorithms is presented in Table 3. As shown in Table 3, in terms of metrics 1 and 4, MOPSO has a better performance than NSGA-III, while in terms of other metrics, NSGA-III outperforms MOPSO.

In addition to the calculated performance measures, the running (CPU) time of the two algorithms are also calculated. The total run time of the NSGA-III and MOPSO algorithms with 100 iterations are obtained as 65.33 and 70.47 s, respectively.

Fig. 8. Final Pareto optimal solutions for the merged MOPSO and NSGA III.

Table 3
Performance metrics summary for NSGA-III and MOPSO.

Metric	Metric 1: Pareto Dominance Indicator	Metric 2: Distribution in diversity metrics	Metric 3: Spacing	Metric 4: Convergence of two sets
NSGA-III	0.4870	-3.54554E-14	94.69776	0.01
MOPSO	0.5130	5.3183E-14	112.9345	0.00

Furthermore, the Pareto optimal solutions of the NSGA-III and MOPSO have been merged and the non-dominated solutions (first layer of the non-dominiated sorting procedure) have been presented as the final Pareto optimal solutions (illustrated in Fig. 8). The first layer consists of seven optimal solutions, DMU40 from MOPSO and 6 DMUs from NSGA-III (DMU08, DMU31, DMU32, DMU57, DMU62 and DMU78). The proportions of the non-dominated solutions in the final Pareto optimal solutions show that NSGA-III is more effective in determining the appropriate solutions than the MOPSO algorithm.

We also applied DEA tools to compare two algorithms by calculating the efficiency of the solutions in terms of the relative efficiency. Furthermore, a popular MCDM tool called TOPSIS is also applied to rank the solutions obtained by the two algorithms. TOPSIS can also be considered as a comparison tool to evaluate the performance of the two algorithms. The following sections present the details concerning the implementation of DEA and TOPSIS.

5.3. Evaluating the efficiency of solutions using DEA

In this section, DEA is performed to compare the relative efficiency of the 100 Pareto optimal solutions obtained from the NSGA-III and MOPSO methods, so a few workable Pareto optimal solutions can be presented for decision making in implementing X-bar economical control charts. When applying the DEA method, all the Pareto optimal solutions can be considered to be DMUs. The cost objective is considered as an input variable, and the average run length and the detection power objectives are considered as output variables. To obtain the efficiencies of the 100 DMUs for each algorithm, a linear program needs to be solved for each DMU. Obviously, as the objective function changes from one linear program to another, and the weights for each DMU may be different. Furthermore, in the DEA method, there may be more than one efficient DMU with relative efficiency equal to one, as each individual DMU is trying to select a preferable weight set when evaluating its efficiency. The higher relative efficiency value means that a higher output value can be obtained for a relatively lower amount of weighted inputs.

Four DEA models, including the input-oriented (io) and output-oriented (oo) models under both the CCR and BCC models, are applied to evaluate the relative efficiency of the 100 solutions from each algorithm. Appendix A and Appendix B show the inputs (cost objective), the outputs (average run length and detection power objectives) and the efficiency results using the four different DEA models. The fully efficient and non-efficient DMUs are represented by (1), and (0), respectively.

As can be observed in Appendix A and Appendix B, 31 solutions out of 100 results obtained from using the NSGA-III algorithm are identified as fully efficient ($\theta_0^* = 1$ *and* $S_i^{-*} = S_r^{+*} = 0$) in all four methods, while only 21 solutions from MOPSO are identified as fully efficient. Table 4 represents the total number of efficient solutions obtained from applying the DEA models for the two optimization algorithms.

The number of fully efficient DMUs in both the BCC input and output oriented models are larger than that from the CCR models, which can be justified by the variable returns to scale assumption of the BCC model with a more flexible frontier selection (Cook & Seiford, 2009). Fig. 9(a) and (b) show the solutions that are recognized as efficient solutions from at least one of the DEA models.

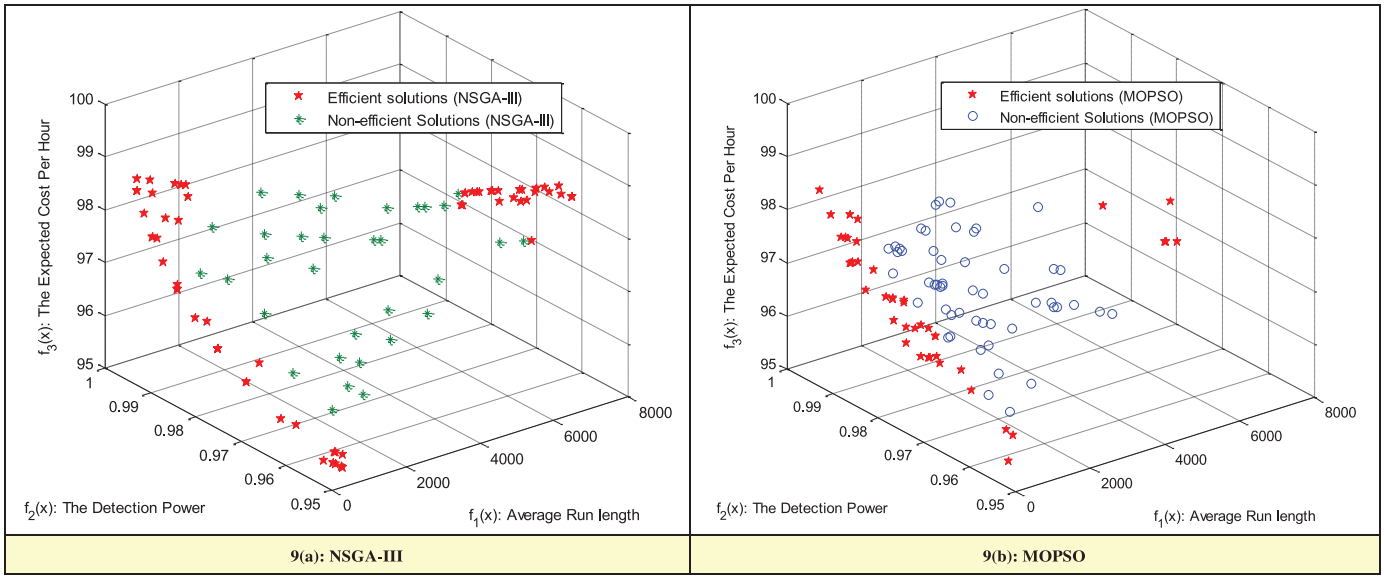


Fig. 9. Efficient versus non-efficient solutions.

Table 4
Number of efficient solutions for NSGA-III and MOPSO obtained from DEA models.

Metric	Total number of efficient solutions			
	CCR (i-o)	CCR (o-o)	BCC (i-o)	BCC (o-o)
NSGA-III	31	31	49	62
MOPSO	21	21	42	36

Table 5
NSGA-III and MOPSO efficient solutions for a set of 200 DMUs.

Metric	Total number of efficient solutions			
	CCR (i-o)	CCR (o-o)	BCC (i-o)	BCC (o-o)
MOPSO	8	8	21	20
NSGA-III	21	21	48	40
Both	29	29	69	60

The solutions from the CCR input and output oriented models are exactly the same for both algorithms, and are plotted in Fig. 10(a) for NSGA-III and in Fig. 10(b) for MOPSO. The efficient units obtained by the BCC input and output oriented models are slightly different, and are plotted in Fig. 10(c) and (e) for NSGA-III and 10(d) and 10(f) for MOPSO.

In summary, the DEA models can significantly reduce a large number of Pareto optimal solutions to a few implementable efficient solutions from an economic perspective that considers all three objectives. The information from both the original Pareto frontier and the efficient solutions from DEA can be used to select the final solutions for the application of the control chart.

The DEA tool is also utilized to compare the performance of the two algorithms in terms of the efficiency of the solutions. For this purpose, all of the solutions are combined to generate 200 DMUs and compared in terms of their efficiency using the four DEA models. A summary of these comparisons is presented in Table 5, which shows a better performance of NSGA-III in generating efficient solutions. Furtheremore, the results show that when the results obtained with both algorithms are combined, the number of efficient solutions increases. Fig. 11(a), (b), and (c) illustrate the efficient solutions when all the results obtained from NSGA-III and MOPSO are considered as a set of 200 DMUs.

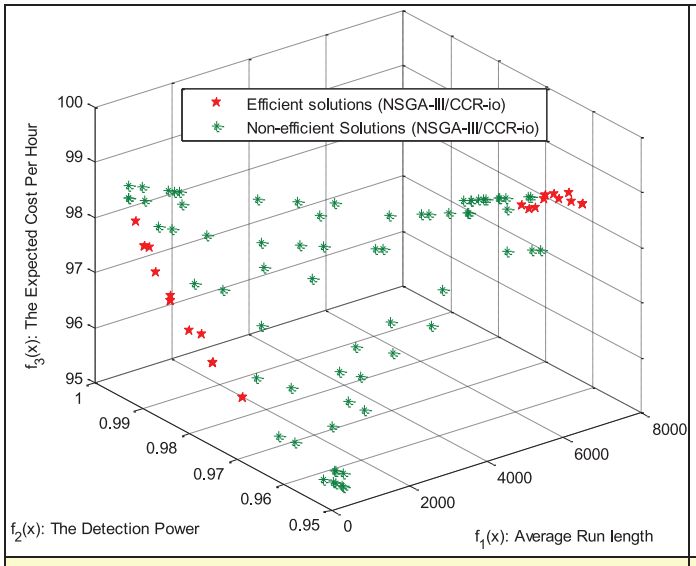
5.4. Efficient solutions prioritization using TOPSIS

In this subsection, the efficient solutions obtained from the four DEA models considering both NSGA-III and MOPSO are prioritized using the TOPSIS technique. For this purpose, objective functions are considered as criteria. The first and second criteria are considered as positive criteria, since the first and second objectives are maximization. The third function is cost, which is considered as a negative criterion. The weights of the criteria are assumed to be equal. After applying the TOPSIS technique, the rank of each DMU is obtained. The top 25 optimal efficient solutions obtained by TOPSIS are presented in Table 6. As can be seen from Table 6, NSGA-III introduces more optimal efficient high-ranked solutions for all the DEA models. Fig. 12(a), (b), and (c) represent the top 25 ranked efficient optimal solutions obtained by NSGA-III and MOPSO, and ranked by the TOPSIS method utilizing the different DEA models.

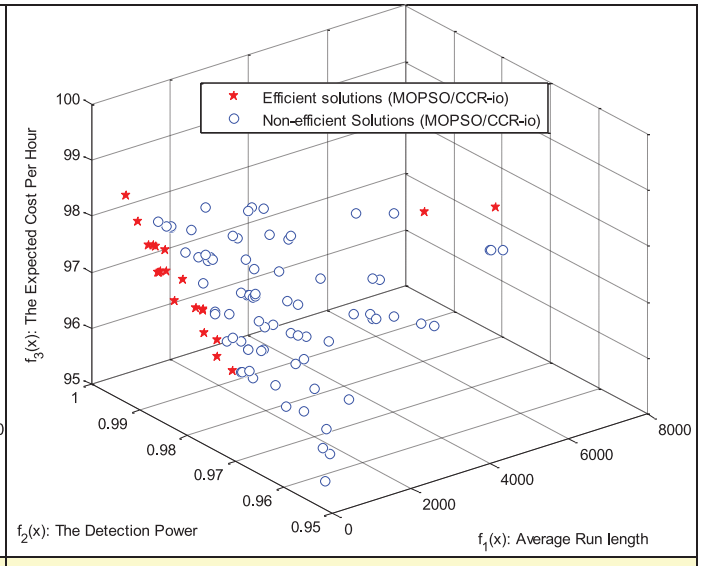
5.5. Solving several developed numerical examples using the proposed optimization method

In order to evaluate the performance of the proposed optimization method, we developed 40 numerical examples, all based on the addressed case study problem, by changing the parameters (a_1, a_2, a_3, a_4, a_5, λ, δ, g, and D), the upper and lower bounds on constraints (α_U and p_L), and decision variable interval (n, h, k). The configurations of the developed problems are presented in Table 7, problem sets A to J. Note that in Table 7, the bold fonts in each problem set indicate the parameters that are different compared to the original case study.

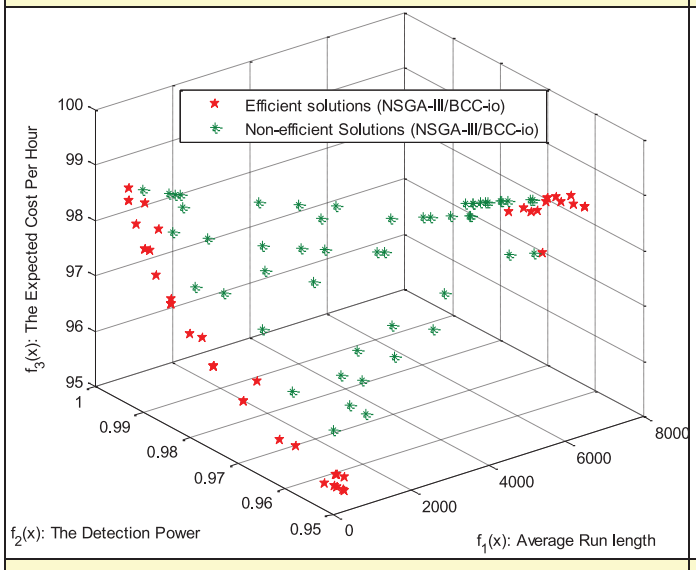
NSGA-III and MOPSO algorithms are utilized to optimize the developed problems. For comparison purposes, the four metrics for Pareto optimal solutions obtained with each algorithm are calculated. In addition, the four DEA methods are utilized to obtain the efficient solutions obtained from NSGA-III and MOPSO. In addition, the TOPSIS method is used to obtain the top-10 high-ranked solutions when all the efficient solutions of NSGA-III and MOPSO are combined. The summary of results is presented in Appendix C. As presented in the last row of Appendix C, NSGA-III outperforms MOPSO in all of the four metrics. For instance, in the metric 1, which measures the number of solutions of the algorithms in the final Pareto frontier after merging the solutions of the algorithms, the percentage of the solutions from NSGA-III is greater than the percentage of the solution



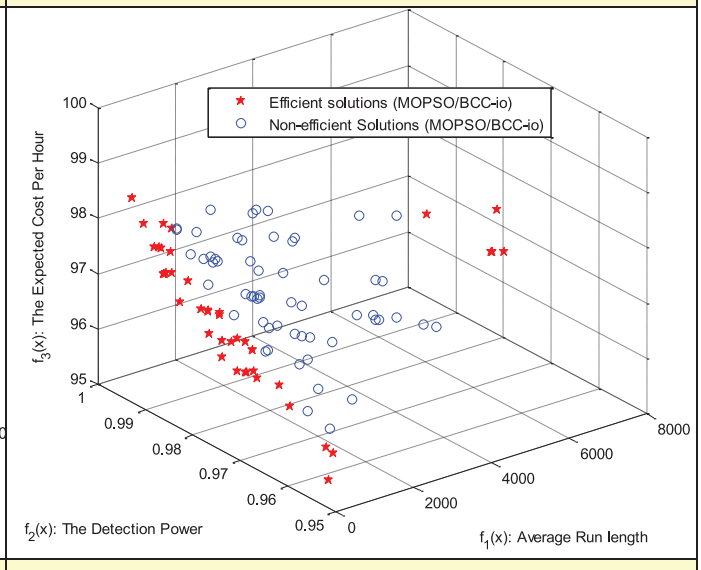
10(a): CCR input-oriented solution for NSGA-III and CCR output-oriented solution for NSGA-III (identical)



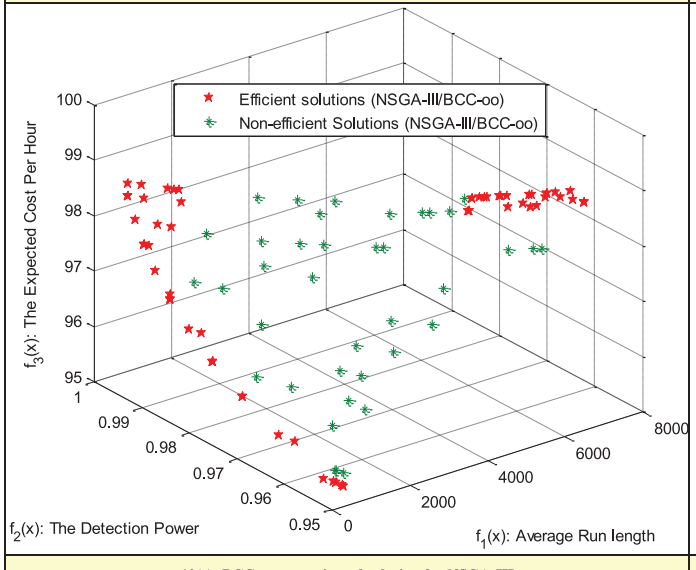
10(b): CCR input-oriented solution for MOPSO and CCR output-oriented solution for MOPSO (identical)



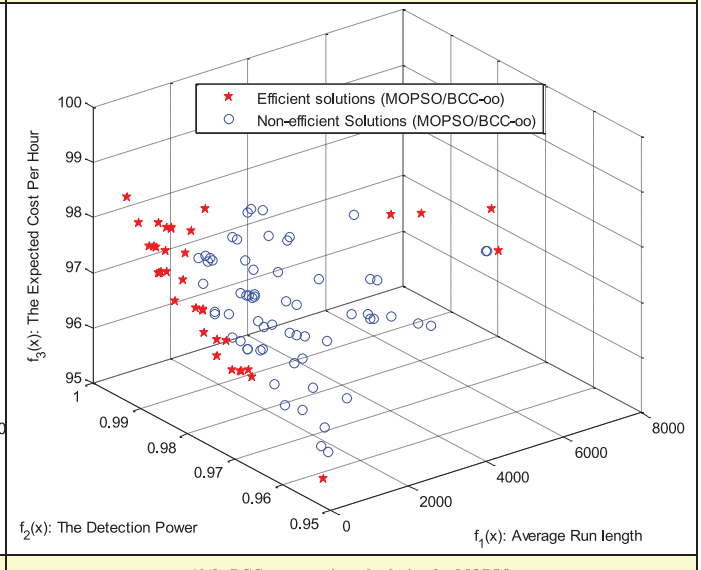
10(c): BCC input-oriented solution for NSGA-III



10(d): BCC input-oriented solution for MOPSO

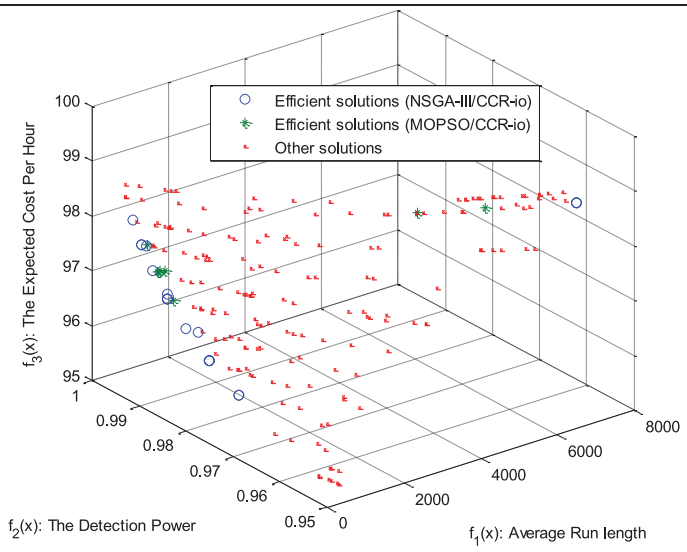


10(e): BCC output-oriented solution for NSGA-III

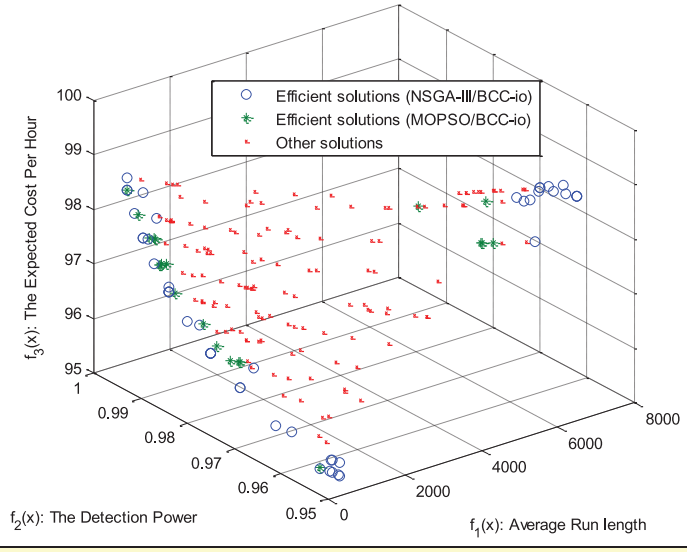


10(f): BCC output-oriented solution for MOPSO

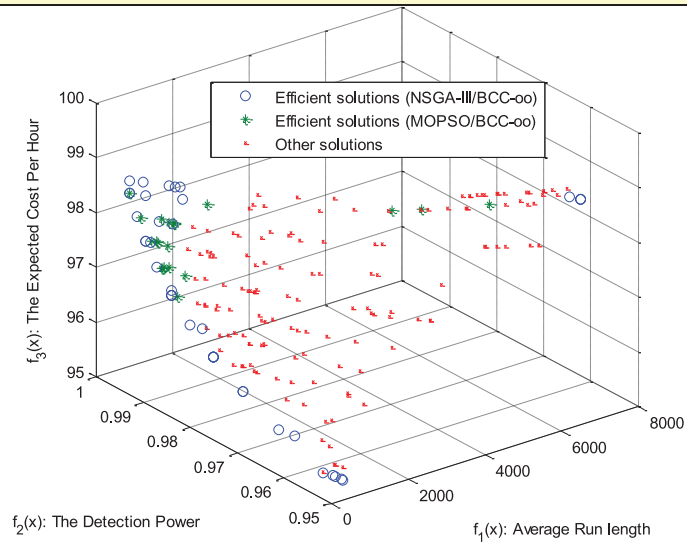
Fig. 10. Efficient versus non-efficient solutions.



11(a): CCR input-oriented solution for NSGA-III and MOPSO and CCR output-oriented solution for NSGA-III and MOPSO (identical)



11(b): BCC input-oriented solution for NSGA-III and MOPSO



11(c): BCC output-oriented solution for NSGA-III and MOPSO

Fig. 11. Efficient versus non-efficient solutions for 200 DMUs.

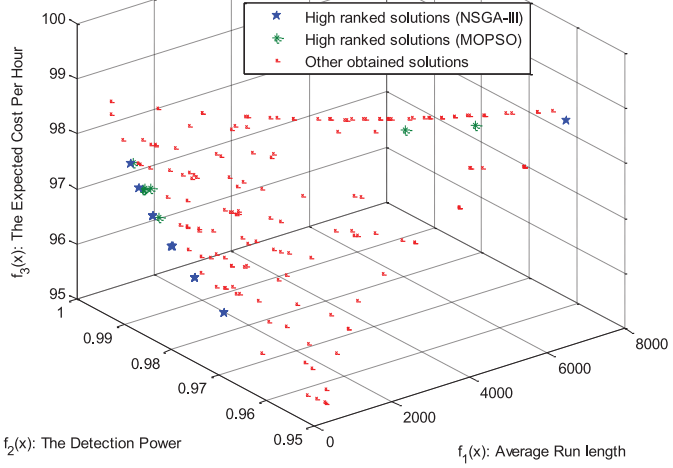
Table 6
Top 25 optimal efficient solutions obtained from NSGA-III/MOPSO/DEA/TOPSIS.

CCR input-oriented and CCR output-oriented rankings (identical)			BCC input-oriented rankings			BCC output-oriented rankings		
Algorithm	DMU number	Rank	Algorithm	DMU number	Rank	Algorithm	DMU number	Rank
NSGA-III	DMU51	1	NSGA-III	DMU51	1	NSGA-III	DMU51	1
NSGA-III	DMU03	2	NSGA-III	DMU03	2	NSGA-III	DMU03	2
NSGA-III	DMU31	3	NSGA-III	DMU31	3	NSGA-III	DMU31	3
NSGA-III	DMU32	4	NSGA-III	DMU32	4	NSGA-III	DMU32	4
MOPSO	DMU28	5	NSGA-III	DMU65	5	NSGA-III	DMU65	5
MOPSO	DMU07	6	NSGA-III	DMU24	6	MOPSO	DMU28	6
MOPSO	DMU76	7	NSGA-III	DMU58	7	MOPSO	DMU07	7
MOPSO	DMU03	8	NSGA-III	DMU23	8	MOPSO	DMU72	8
NSGA-III	DMU90	9	NSGA-III	DMU52	9	MOPSO	DMU64	9
NSGA-III	DMU10	10	NSGA-III	DMU92	10	NSGA-III	DMU94	10
NSGA-III	DMU47	11	NSGA-III	DMU30	11	NSGA-III	DMU45	11
NSGA-III	DMU53	12	NSGA-III	DMU67	12	NSGA-III	DMU05	12
MOPSO	DMU34	12	NSGA-III	DMU35	12	NSGA-III	DMU43	13
MOPSO	DMU37	12	NSGA-III	DMU60	12	MOPSO	DMU13	14
MOPSO	DMU61	15	NSGA-III	DMU25	15	MOPSO	DMU50	15
NSGA-III	DMU12	16	MOPSO	DMU28	16	NSGA-III	DMU74	16
MOPSO	DMU22	17	MOPSO	DMU87	17	MOPSO	DMU25	17
NSGA-III	DMU85	18	MOPSO	DMU65	18	MOPSO	DMU92	18
NSGA-III	DMU13	19	MOPSO	DMU23	19	NSGA-III	DMU68	19
NSGA-III	DMU17	20	MOPSO	DMU07	20	MOPSO	DMU90	20
NSGA-III	DMU19	21	NSGA-III	DMU68	21	NSGA-III	DMU83	21
NSGA-III	DMU57	22	NSGA-III	DMU83	22	MOPSO	DMU31	22
NSGA-III	DMU14	23	NSGA-III	DMU09	23	NSGA-III	DMU44	23
NSGA-III	DMU15	24	MOPSO	DMU04	24	MOPSO	DMU04	24
NSGA-III	DMU78	25	MOPSO	DMU91	25	MOPSO	DMU91	25

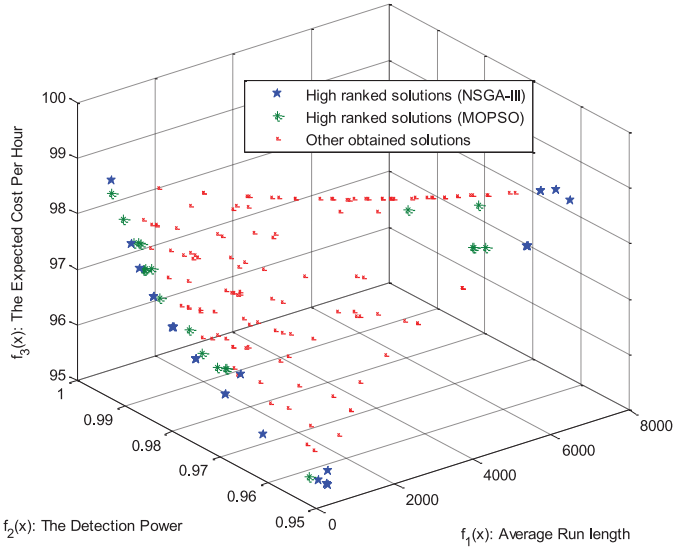
Table 7
Parameters of the developed numerical examples*.

Instances	Problem #	Model parameters									Model constraints		Decision variables interval		
Problem set		a_1	a_2	a_3	a_4	a_5	λ	g	δ	D	α_U	p_L	n	h	K
A	A1	**0.5**	**0.05**	**30**	**30**	**100**	0.25	0.010	1.0	2.0	0.005	0.950	[20,30]	[0.40,0.50]	[2.90,3.80]
	A2	**0.8**	**0.08**	**40**	**40**	**150**	0.25	0.010	1.0	2.0	0.005	0.950	[20,30]	[0.40,0.50]	[2.90,3.80]
	A3	**10**	**1.0**	**80**	**80**	**350**	0.25	0.010	1.0	2.0	0.005	0.950	[20,30]	[0.40,0.50]	[2.90,3.80]
	A4	**20**	**2.0**	**100**	**100**	**500**	0.25	0.010	1.0	2.0	0.005	0.950	[20,30]	[0.40,0.50]	[2.90,3.80]
B	B1	1.0	0. 1	50	50	200	**0.05**	0.010	1.0	2.0	0.005	0.950	[20,30]	[0.40,0.50]	[2.90,3.80]
	B2	1.0	0. 1	50	50	200	**0.15**	0.010	1.0	2.0	0.005	0.950	[20,30]	[0.40,0.50]	[2.90,3.80]
	B3	1.0	0. 1	50	50	200	**0.35**	0.010	1.0	2.0	0.005	0.950	[20,30]	[0.40,0.50]	[2.90,3.80]
	B4	1.0	0. 1	50	50	200	**0.45**	0.010	1.0	2.0	0.005	0.950	[20,30]	[0.40,0.50]	[2.90,3.80]
C	C1	1.0	0. 1	50	50	200	0.25	**0.005**	1.0	2.0	0.005	0.950	[20,30]	[0.40,0.50]	[2.90,3.80]
	C2	1.0	0. 1	50	50	200	0.25	**0.008**	1.0	2.0	0.005	0.950	[20,30]	[0.40,0.50]	[2.90,3.80]
	C3	1.0	0. 1	50	50	200	0.25	**0.080**	1.0	2.0	0.005	0.950	[20,30]	[0.40,0.50]	[2.90,3.80]
	C4	1.0	0. 1	50	50	200	0.25	**0.100**	1.0	2.0	0.005	0.950	[20,30]	[0.40,0.50]	[2.90,3.80]
D	D1	1.0	0. 1	50	50	200	0.25	0.010	**0.2**	2.0	0.005	0.950	[20,30]	[0.40,0.50]	[2.90,3.80]
	D2	1.0	0. 1	50	50	200	0.25	0.010	**0.5**	2.0	0.005	0.950	[20,30]	[0.40,0.50]	[2.90,3.80]
	D3	1.0	0. 1	50	50	200	0.25	0.010	**1.2**	2.0	0.005	0.950	[20,30]	[0.40,0.50]	[2.90,3.80]
	D4	1.0	0. 1	50	50	200	0.25	0.010	**1.5**	2.0	0.005	0.950	[20,30]	[0.40,0.50]	[2.90,3.80]
E	E1	1.0	0. 1	50	50	200	0.25	0.010	1.0	**0.5**	0.005	0.950	[20,30]	[0.40,0.50]	[2.90,3.80]
	E2	1.0	0. 1	50	50	200	0.25	0.010	1.0	**1.0**	0.005	0.950	[20,30]	[0.40,0.50]	[2.90,3.80]
	E3	1.0	0. 1	50	50	200	0.25	0.010	1.0	**3.0**	0.005	0.950	[20,30]	[0.40,0.50]	[2.90,3.80]
	E4	1.0	0. 1	50	50	200	0.25	0.010	1.0	**5.0**	0.005	0.950	[20,30]	[0.40,0.50]	[2.90,3.80]
F	F1	1.0	0. 1	50	50	200	0.25	0.010	1.0	2.0	**0.001**	0.950	[20,30]	[0.40,0.50]	[2.90,3.80]
	F2	1.0	0. 1	50	50	200	0.25	0.010	1.0	2.0	**0.003**	0.950	[20,30]	[0.40,0.50]	[2.90,3.80]
	F3	1.0	0. 1	50	50	200	0.25	0.010	1.0	2.0	**0.010**	0.950	[20,30]	[0.40,0.50]	[2.90,3.80]
	F4	1.0	0. 1	50	50	200	0.25	0.010	1.0	2.0	**0.030**	0.950	[20,30]	[0.40,0.50]	[2.90,3.80]
G	G1	1.0	0. 1	50	50	200	0.25	0.010	1.0	2.0	0.005	**0.910**	[20,30]	[0.40,0.50]	[2.90,3.80]
	G2	1.0	0. 1	50	50	200	0.25	0.010	1.0	2.0	0.005	**0.930**	[20,30]	[0.40,0.50]	[2.90,3.80]
	G3	1.0	0. 1	50	50	200	0.25	0.010	1.0	2.0	0.005	**0.970**	[20,30]	[0.40,0.50]	[2.90,3.80]
	G4	1.0	0. 1	50	50	200	0.25	0.010	1.0	2.0	0.005	**0.990**	[20,30]	[0.40,0.50]	[2.90,3.80]
H	H1	1.0	0. 1	50	50	200	0.25	0.010	1.0	2.0	0.005	0.950	**[14,24]**	[0.40,0.50]	[2.90,3.80]
	H2	1.0	0. 1	50	50	200	0.25	0.010	1.0	2.0	0.005	0.950	**[17,27]**	[0.40,0.50]	[2.90,3.80]
	H3	1.0	0. 1	50	50	200	0.25	0.010	1.0	2.0	0.005	0.950	**[23,33]**	[0.40,0.50]	[2.90,3.80]
	H4	1.0	0. 1	50	50	200	0.25	0.010	1.0	2.0	0.005	0.950	**[26,36]**	[0.40,0.50]	[2.90,3.80]
I	I1	1.0	0. 1	50	50	200	0.25	0.010	1.0	2.0	0.005	0.950	[20,30]	**[0.20,0.30]**	[2.90,3.80]
	I2	1.0	0. 1	50	50	200	0.25	0.010	1.0	2.0	0.005	0.950	[20,30]	**[0.30,0.40]**	[2.90,3.80]
	I3	1.0	0. 1	50	50	200	0.25	0.010	1.0	2.0	0.005	0.950	[20,30]	**[0.50,0.60]**	[2.90,3.80]
	I4	1.0	0. 1	50	50	200	0.25	0.010	1.0	2.0	0.005	0.950	[20,30]	**[0.60,0.70]**	[2.90,3.80]
J	J1	1.0	0. 1	50	50	200	0.25	0.010	1.0	2.0	0.005	0.950	[20,30]	[0.40,0.50]	**[2.70,3.60]**
	J2	1.0	0. 1	50	50	200	0.25	0.010	1.0	2.0	0.005	0.950	[20,30]	[0.40,0.50]	**[2.80,3.70]**
	J3	1.0	0. 1	50	50	200	0.25	0.010	1.0	2.0	0.005	0.950	[20,30]	[0.40,0.50]	**[3.00,3.90]**
	J4	1.0	0. 1	50	50	200	0.25	0.010	1.0	2.0	0.005	0.950	[20,30]	[0.40,0.50]	**[3.10,4.00]**

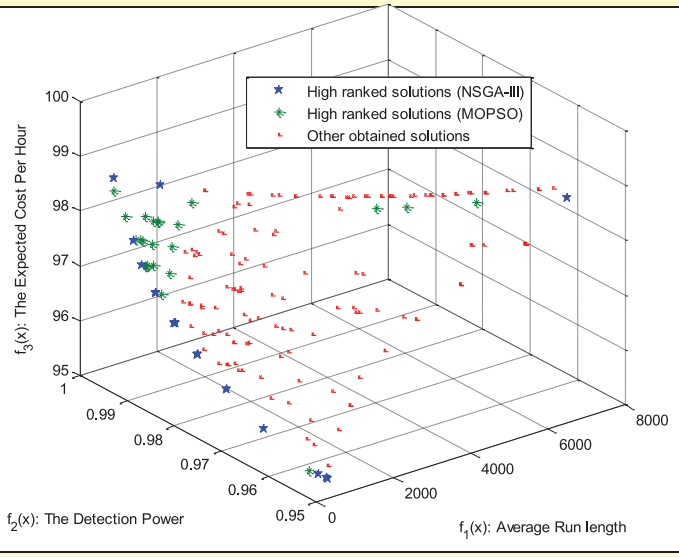
* *Note*: The bold fonts in each problem set indicate the parameters that are different compared to the original case study.



12(a): CCR input-oriented solution for NSGA-III and MOPSO and CCR output-oriented solution for NSGA-III and MOPSO (identical)



12(b): BCC input-oriented solution for NSGA-III and MOPSO



12(c): BCC output-oriented solution for NSGA-III and MOPSO

Fig. 12. Top 25 efficient solutions obtained by the NSGA-III/MOPSO/TOPSIS methods.

of MOPSO. Also, in terms of distribution and spacing metrics, NSGA-III has a smaller metric value, which indicates that NSGA-III is better than MOPSO. Considering the number of efficient optimal solutions, NSGA-III outperforms MOPSO in all 40 numerical examples. Furthermore, prioritizing the combined efficient solutions using TOPSIS reveals that the number of high-ranked efficient solutions obtained by NSGA-III is higher than MOPSO in all numerical examples.

6. Conclusion

In this study, we compared the performance of two multi-objective optimization algorithms, including NSGA-III and MOPSO, to optimize an X-bar economical control chart design problem. Both algorithms are modified to consider three objectives and two constraints with discrete and continuous decision variables. After obtaining 100 optimal solutions from each algorithm, different metrics were calculated to compare the Pareto optimal frontiers obtained from using each algorithm. The performance measurements revealed that NSGA-III attains better outcomes for the majority of the calculated metrics. In addition, all 200 solutions obtained by the two algorithms were merged to find a final Pareto frontier, which includes all the non-dominated solutions. We conclude that the NSGA-III algorithm generates more non-dominated solutions than the MOPSO algorithm.

We also used four different input- and output-oriented DEA models to compare 100 optimal solutions founded by each algorithm in terms of the relative efficiency. Moreover, the solutions obtained by each algorithm were combined and then DEA was used to identify the efficient solutions from these 200 solutions. The comparison of the number of efficient solutions revealed a better performance for the NSGA-III algorithm for all DEA models. Furthermore, by using the relative efficiency concepts of DEA for eliminating the inefficient Pareto optimal solutions, the number of solutions was reduced to make the solution space more manageable and improve decision making efficiency and effectiveness.

However, even after reducing the number of optimal solutions using DEA, selecting the best design remained somewhat difficult for the decision makers who were still facing many optimal efficient solutions with identical efficiencies. Therefore, TOPSIS was used to rank the optimal efficient solution obtained by each algorithm. The TOPSIS ranking process provided a good comparison of the two algorithms and showed that the top-ranked efficient solutions are mostly generated by the NSGA-III algorithm.

It should be mentioned that the solutions we obtained are better than the solutions reported in the literature because we: (1) explored the entire feasible solution space by utilizing two efficient multi-objective optimization algorithms; (2) utilized different DEA models to find the efficient optimal solutions; and (3) used TOPSIS to further screen provided the high-ranked solutions for a more effective and efficient decision making process.

We also tested tested forty simulated numerical examples based on the case study problem and utilized the proposed algorithm to first validate its performance, and second, to compare the performance of the NSGA-III and MOPSO algorithms using different metrics. The results obtained from these investigation shows that the NSGA-III algorithm outperforms the MOPSO algorithm.

We should note that, the number of feasible solutions in the framework proposed in this study is not limited because the multi-objective problem addressed in this study includes both continuous and discrete variables. Therefore, using a brute-force search algorithm may not be an efficient alternative solution strategy since exploring and checking all possible solutions could take a great deal of computational time. The metaheuristic algorithms applied in this paper have a well-defined structure and search strategy for exploring the search area (feasible solution region) within a limited number of investigations and in a reasonable computational time.

For future research, other multi-objective optimization algorithms, such as the non-dominated ranked genetic algorithm (NRGA) can be used to optimize different control chart design problems (i.e., R and S charts). The proposed NSGA III approach has been shown to be efficient and therefore applying it to other multi-objective problems can be an interesting and rewarding future research topic.

Acknowledgment

The authors would like to thank the anonymous reviewers and the editor for their insightful comments and suggestions.

Appendix A. Inputs, output, and results of the CCR and BCC models for NSGA-III

DMUs	Decision variables			Output 1	Output 2	Input	CCR	CCR	BCC	BCC
	n	h	k	$f_1 = ARL_0(s)$	$f_2 = p(s)$	$f_3 = E_{HC}(s)$	(I-O)	(O-O)	(I-O)	(O-O)
1	30	0.4	3.71639	4948.293	0.960867	99.0269	0	0	0	1
2	30	0.400001	3.762355	5941.029	0.956816	99.05593	0	0	0	1
3	30	0.465658	3.8	6911.037	0.953251	98.86286	1	1	1	1
4	21	0.48639	2.937626	302.3558	0.95001	95.59464	0	0	1	0
5	30	0.4	3.193579	711.6377	0.988804	98.83282	0	0	0	1
6	21	0.408696	2.934483	299.308	0.950333	95.34947	0	0	1	1
7	21	0.427263	2.919343	285.0867	0.951867	95.35428	0	0	1	1
8	30	0.404579	2.9	267.9797	0.99502	98.75773	0	0	1	1
9	24	0.437024	3.031193	410.527	0.969104	96.42414	0	0	1	0
10	28	0.447659	2.931126	296.0898	0.990872	97.81805	1	1	1	1
11	22	0.420365	2.9	267.9797	0.963306	95.66232	0	0	1	1
12	27	0.456291	2.9	267.9797	0.989166	97.44087	1	1	1	1
13	26	0.448072	2.9	267.9797	0.986062	97.07905	1	1	1	1
14	25	0.44065	2.9	267.9797	0.982136	96.71889	1	1	1	1
15	24	0.43329	2.9	267.9797	0.977195	96.36158	1	1	1	1
16	23	0.425484	2.9	267.9797	0.971009	96.00862	1	1	1	1
17	26	0.448072	2.9	267.9797	0.986062	97.07905	1	1	1	1
18	24	0.437024	2.900012	267.9901	0.977194	96.3622	1	1	1	1
19	26	0.448072	2.9	267.9797	0.986062	97.07905	1	1	1	1
20	24	0.439658	2.900046	268.0187	0.977192	96.36335	1	1	1	1
21	30	0.4	3.71639	4948.293	0.960867	99.0269	0	0	0	1
22	30	0.4	3.74008	5435.913	0.958819	99.04155	0	0	0	1
23	30	0.4	3.780296	6383.972	0.955145	99.06797	1	1	1	1
24	30	0.4	3.790708	6656.966	0.954152	99.07514	1	1	1	1
25	29	0.453002	3.739698	5427.663	0.950063	98.51583	0	0	1	0
26	30	0.4	3.727311	5167.052	0.959934	99.03357	0	0	0	1
27	30	0.4	3.722529	5070.028	0.960344	99.03063	0	0	0	1
28	30	0.4	3.73874	5407.012	0.958937	99.0407	0	0	0	1
29	30	0.4	3.763663	5972.21	0.956695	99.05679	0	0	0	1
30	30	0.410139	3.773796	6219.604	0.955756	98.99929	1	1	1	1
31	30	0.463201	3.8	6911.037	0.953251	98.86371	1	1	1	1
32	30	0.463201	3.8	6911.037	0.953251	98.86371	1	1	1	1
33	30	0.4	3.728875	5199.212	0.959798	99.03454	0	0	0	1
34	30	0.4	3.745467	5553.739	0.958342	99.04497	0	0	0	1
35	30	0.471039	3.762931	5954.729	0.956763	98.83351	1	1	1	1
36	30	0.46302	3.712905	4880.574	0.961161	98.79935	0	0	0	1
43	30	0.4	3.166253	647.5988	0.989583	98.82756	0	0	1	1
44	30	0.400418	3.017969	392.9663	0.993039	98.80116	0	0	0	1
45	30	0.4	3.213226	761.8879	0.988213	98.83682	0	0	0	1
46	30	0.465821	2.9	267.9797	0.99502	98.53164	0	0	1	1
47	28	0.462518	2.9	267.9797	0.99161	97.80332	1	1	1	1
48	21	0.408877	2.9	267.9797	0.953771	95.32335	0	0	1	1
49	21	0.407745	2.937691	302.4197	0.950003	95.35195	0	0	1	1
50	30	0.477689	2.9	267.9797	0.99502	98.52725	0	0	1	1
51	30	0.468571	3.8	6911.037	0.953251	98.86247	1	1	1	1
52	30	0.4	3.77422	6230.168	0.955717	99.06384	1	1	1	1
53	28	0.447101	2.9	267.9797	0.99161	97.81324	1	1	1	1
54	22	0.419634	2.937728	302.4558	0.960172	95.68596	0	0	1	1
56	30	0.446406	3.745768	5560.411	0.958315	98.8431	0	0	1	1
57	26	0.403434	2.9	267.9797	0.986062	97.17097	1	1	1	1
58	30	0.416999	3.784322	6488.119	0.954763	98.9702	1	1	1	1
60	30	0.429578	3.757437	5825.371	0.957265	98.89788	1	1	1	1
62	23	0.429109	2.9	267.9797	0.971009	96.00927	1	1	1	1
64	30	0.462723	3.714689	4915.126	0.961011	98.80074	0	0	0	1
65	30	0.432301	3.792202	6697.149	0.954008	98.91357	1	1	1	1
67	30	0.463767	3.767254	6058.667	0.956364	98.83795	1	1	1	1
68	29	0.480439	3.054246	443.209	0.990121	98.19682	0	0	1	1
74	29	0.451435	3.11981	552.5847	0.988255	98.21852	0	0	0	1
78	24	0.435121	2.9	267.9797	0.977195	96.36174	1	1	1	1
80	21	0.484076	2.923703	289.1055	0.951429	95.56921	0	0	1	0
83	30	0.455562	3.028327	406.6492	0.992835	98.56094	0	0	1	1
85	25	0.453754	2.946943	311.5923	0.979966	96.74243	1	1	1	1
86	21	0.425844	2.922966	288.4216	0.951504	95.35484	0	0	1	1
90	29	0.477689	2.9	267.9797	0.993525	98.16804	1	1	1	1
91	21	0.484709	2.920022	285.7092	0.951799	95.56946	0	0	1	0
92	30	0.4	3.77422	6230.168	0.955717	99.06384	1	1	1	1
94	30	0.440591	3.223934	790.8735	0.98788	98.63219	0	0	0	1

Appendix B. Inputs, output, and results of the CCR and BCC models MOPSO

DMUs	Decision variables			Output 1	Output 2	Input	CCR	CCR	BCC	BCC
	n	h	K	$f_1 = ARL_0(s)$	$f_2 = p(s)$	$f_3 = E_{HC}(s)$	(I-O)	(O-O)	(I-O)	(O-O)
3	28	0.493337	2.941219	305.8822	0.99062	97.84566	1	1	1	1
4	28	0.42983	2.981426	348.5425	0.989558	97.86387	1	1	1	1
6	25	0.441988	3.09428	506.865	0.971657	96.79912	0	0	1	0
7	30	0.461157	3.67073	4134.645	0.96458	98.77313	1	1	1	1
10	23	0.489157	3.012972	386.5441	0.962695	96.23554	0	0	1	0
13	29	0.487094	3.128476	569.1017	0.987986	98.22112	0	0	0	1
15	26	0.449251	3.079052	481.5548	0.978307	97.13847	0	0	1	0
19	25	0.441415	3.00678	378.7442	0.976881	96.75886	1	1	1	1
22	26	0.462764	2.929019	294.0886	0.984997	97.09556	1	1	1	1
23	29	0.461852	3.695318	4553.489	0.954471	98.47644	0	0	1	0
25	29	0.446575	3.104093	523.9352	0.988728	98.2229	0	0	1	1
26	26	0.4624	3.041308	424.5336	0.980191	97.13284	1	1	1	1
27	24	0.434236	3.07894	481.3746	0.965623	96.45003	0	0	1	0
28	30	0.437344	3.733528	5296.156	0.959394	98.85651	1	1	1	1
30	25	0.40944	3.034441	414.9683	0.975325	96.81219	0	0	1	1
31	28	0.480909	3.031425	410.8425	0.988092	97.84371	1	1	1	1
32	25	0.405607	2.963213	328.4658	0.979164	96.79661	1	1	1	1
34	27	0.486008	2.939243	303.9377	0.987993	97.48484	1	1	1	1
35	25	0.439363	3.094831	507.8075	0.971621	96.79904	0	0	1	0
36	24	0.478745	3.012946	386.5106	0.970355	96.49836	0	0	1	1
37	27	0.439409	2.932375	297.2827	0.988206	97.45889	1	1	1	1
38	25	0.499747	3.05291	441.2396	0.974238	96.9132	0	0	1	0
40	21	0.424215	2.9	267.9797	0.953771	95.33462	0	0	1	1
42	26	0.424918	3.038501	420.5952	0.980325	97.1446	1	1	1	1
44	26	0.412383	3.080012	483.1106	0.978257	97.19224	0	0	1	0
45	29	0.441029	3.205243	741.0231	0.985368	98.25944	0	0	0	1
50	29	0.495956	3.125668	563.693	0.988074	98.23492	0	0	0	1
57	29	0.473674	2.937271	302.0104	0.992815	98.17176	1	1	1	1
59	30	0.47784	2.900047	268.0203	0.995019	98.52726	1	1	1	1
61	27	0.436374	2.926097	291.3383	0.988398	97.46202	1	1	1	1
64	30	0.480948	3.307119	1060.888	0.985001	98.60709	0	0	0	1
65	29	0.461057	3.696473	4574.246	0.954361	98.47742	0	0	1	0
67	24	0.429661	3.022482	398.866	0.969706	96.4186	0	0	1	1
71	25	0.495719	3.077573	479.1717	0.972724	96.90966	0	0	1	0
72	30	0.474055	3.639382	3659.078	0.966957	98.75028	0	0	0	1
75	26	0.449372	3.013024	386.6101	0.98151	97.11385	1	1	1	1
76	27	0.49857	2.96401	329.3176	0.987197	97.5243	1	1	1	1
83	24	0.452339	2.988742	356.9784	0.971949	96.41784	0	0	1	1
84	24	0.437644	2.965795	331.2339	0.973393	96.39143	1	1	1	1
86	28	0.423975	3.119682	552.3453	0.985065	97.91455	0	0	0	1
87	29	0.466216	3.706191	4752.865	0.953421	98.48565	0	0	1	1
88	24	0.456817	2.992485	361.3812	0.971707	96.42773	0	0	1	1
90	27	0.460335	3.04206	425.5955	0.984384	97.47869	1	1	1	1
91	28	0.428277	2.973137	339.2444	0.989785	97.86679	1	1	1	1
92	29	0.429434	3.066377	461.5312	0.989797	98.26063	0	0	1	1
93	22	0.470054	2.995309	364.7412	0.954972	95.84487	0	0	1	0
94	22	0.454548	3.010011	382.7922	0.953561	95.80019	0	0	1	0
97	24	0.49852	2.919578	285.3019	0.976115	96.52921	1	1	1	1

Appendix C. Summary of results obtained by solving developed problems

Problem title	Method	Objective functions									Metrics				Total # of efficient solutions				# of high-ranked solutions (out of 10) using TOPSIS			
		$ARL_0(s)$			p(s)			$E_{HC}(s)$															
		min	max	Ave.	min	max	Ave.	min	max	Ave.	1	2	3	4	CCR (io)	CCR (oo)	BCC (io)	BCC (oo)	CCR (io)	CCR (oo)	BCC (io)	BCC (oo)
A1	NSGA-III	267.98	6911.04	2458.15	0.9500	0.9929	0.9647	48.46	50.32	49.70	0.500	−2.E−14	5.6402	0.00	20	20	33	39	6	6	8	9
	MOPSO	267.98	6243.91	1263.08	0.9503	0.9945	0.9716	48.59	50.18	49.60	0.500	−1.E−14	15.087	0.00	15	15	30	31	4	4	2	1
A2	NSGA-III	267.98	6911.04	2471.77	0.9500	0.9948	0.9651	72.26	75.20	74.21	0.500	9.E−15	5.6392	0.00	21	21	35	38	7	7	9	10
	MOPSO	271.59	5771.19	1465.04	0.9502	0.9925	0.9711	72.61	75.15	74.11	0.500	−5.E-14	4.2501	0.00	15	15	32	30	3	3	1	0
A3	NSGA-III	267.98	6911.04	2897.50	0.9500	0.9948	0.9664	217.42	254.70	240.17	0.709	-4.E−14	4.1553	0.00	20	20	36	39	7	7	7	8
	MOPSO	286.99	5224.24	1242.78	0.9507	0.9895	0.9717	222.77	250.24	237.12	0.291	9.E−15	5.8381	0.26	14	14	31	29	3	3	3	2
A4	NSGA-III	267.98	6911.04	2956.84	0.9500	0.9949	0.9665	343.77	418.79	389.65	0.719	−1.E−14	5.2752	0.00	20	20	34	34	6	6	7	7
	MOPSO	267.98	6782.52	1242.28	0.9518	0.9935	0.9753	356.83	413.95	385.76	0.281	−2.E−14	14.847	0.32	16	16	30	29	4	4	3	3
B1	NSGA-III	267.98	6911.04	2803.95	0.9500	0.9942	0.9658	32.95	36.72	35.32	0.649	−1.E-16	5.5271	0.00	23	23	36	39	7	7	8	8
	MOPSO	268.80	6911.04	1162.19	0.9507	0.9918	0.9736	33.31	36.49	35.07	0.351	−3.E−14	25.085	0.17	17	17	31	30	3	3	2	2
B2	NSGA-III	267.98	6911.04	2497.33	0.9500	0.9947	0.9652	70.23	73.87	72.63	0.505	−4.E−14	4.4889	0.00	25	25	35	38	7	7	8	9
	MOPSO	267.98	6490.28	1266.25	0.9501	0.9946	0.9738	70.57	73.74	72.49	0.495	−2.E−14	12.351	0.03	16	16	30	33	3	3	2	1
B3	NSGA-III	267.98	6911.04	2430.77	0.9500	0.9950	0.9647	113.54	117.28	116.06	0.498	−2.E−14	5.7041	0.01	24	24	38	43	6	6	9	10
	MOPSO	267.98	6345.90	924.481	0.9517	0.9950	0.9739	113.53	117.11	115.72	0.503	−6.E−14	16.811	0.00	15	15	26	34	4	4	1	0
B4	NSGA-III	267.98	6911.04	2428.39	0.9500	0.9950	0.9647	127.35	131.04	129.83	0.500	−5.E−15	5.9478	0.00	26	26	39	40	7	7	8	9
	MOPSO	272.75	5408.48	1134.15	0.9500	0.9938	0.9731	127.70	130.87	129.65	0.500	2.E−14	7.2794	0.00	17	17	32	30	3	3	2	1
C1	NSGA-III	267.98	6911.04	2415.95	0.9500	0.9950	0.9649	93.52	96.53	95.53	0.500	7.E−15	5.7110	0.00	25	25	36	45	6	6	8	8
	MOPSO	276.18	5253.70	1188.96	0.9509	0.9925	0.9732	93.53	96.40	95.41	0.500	−4.E−15	6.8304	0.00	20	20	30	32	4	4	2	2
C2	NSGA-III	267.98	6911.04	2418.66	0.9500	0.9950	0.9647	94.63	98.08	96.93	0.500	3.E−14	4.8453	0.00	29	29	38	48	6	6	7	7
	MOPSO	271.16	5856.65	1184.75	0.9503	0.9936	0.9736	94.96	97.94	96.77	0.500	1.E−14	12.086	0.00	21	21	30	36	4	4	3	3
C3	NSGA-III	267.98	6911.04	2430.06	0.9500	0.9950	0.9652	115.39	125.42	122.13	0.503	7.E−15	6.9789	0.00	30	30	40	61	7	7	8	8
	MOPSO	267.98	6350.92	1382.25	0.9505	0.9935	0.9726	116.47	125.20	122.07	0.498	1.E−14	7.3155	0.04	22	22	32	35	3	3	2	2
C4	NSGA-III	267.98	6911.04	2440.33	0.9500	0.9928	0.9646	119.78	130.81	127.17	0.503	4.E−14	6.1299	0.00	28	28	39	60	6	6	7	8
	MOPSO	267.98	4713.96	1198.50	0.9508	0.9947	0.9745	119.86	130.53	127.23	0.498	9.E−15	4.5022	0.01	20	20	33	35	4	4	3	2
D1	NSGA-III	302.53	6897.27	3102.67	0.0034	0.0327	0.0114	173.37	203.71	193.65	0.769	3.E−15	4.1491	0.00	26	26	36	50	8	8	9	10
	MOPSO	267.98	6596.39	1058.80	0.0035	0.0335	0.0165	170.84	203.04	186.41	0.231	−9.E−15	23.092	0.55	12	12	18	35	2	2	1	0
D2	NSGA-III	525.50	6911.04	3603.95	0.1443	0.3570	0.2031	111.26	131.06	122.60	0.769	−8.E−15	4.2491	0.00	28	28	38	45	7	7	9	10
	MOPSO	272.04	5882.09	1143.42	0.1536	0.4136	0.2698	107.35	128.82	115.15	0.231	1.E−14	9.7814	0.64	10	10	15	18	3	3	1	0
D3	NSGA-III	267.98	6911.04	2496.28	0.9948	0.9998	0.9971	95.02	98.78	97.21	0.500	−8.E−15	4.1364	0.00	25	25	40	48	8	8	8	9
	MOPSO	269.45	4857.84	1281.34	0.9954	0.9999	0.9984	95.44	98.64	97.60	0.500	−2.E−15	5.9763	0.00	12	12	18	20	2	2	2	1
D4	NSGA-III	267.98	6911.04	2280.09	0.9999	0.9999	0.9999	94.99	98.76	96.70	0.505	−2.E−14	4.0674	0.00	24	24	41	50	7	7	8	8
	MOPSO	267.98	4348.70	1020.57	0.9999	0.9999	0.9999	95.41	98.73	97.41	0.495	−1.E−14	12.694	0.02	8	8	17	29	3	3	2	2
E1	NSGA-III	267.98	6911.04	2626.46	0.9500	0.9949	0.9651	62.58	68.34	66.34	0.585	1.E−14	5.1143	0.00	27	27	40	48	7	7	8	9
	MOPSO	267.98	6567.33	1218.95	0.9511	0.9950	0.9731	63.76	68.15	66.15	0.415	−1.E−14	11.096	0.12	15	15	31	30	3	3	2	1
E2	NSGA-III	267.98	6911.04	2502.08	0.9500	0.9950	0.9649	75.43	80.22	78.61	0.521	−9.E−15	5.0305	0.00	26	26	42	58	7	7	8	9
	MOPSO	271.88	6036.92	1111.74	0.9509	0.9936	0.9746	75.59	80.09	78.42	0.479	−8.E−16	6.3048	0.02	12	12	29	30	3	3	2	1
E3	NSGA-III	267.98	6911.04	2596.62	0.9500	0.9950	0.9652	109.85	113.41	112.18	0.513	−2.E−14	5.6949	0.00	28	28	45	52	8	8	9	1
	MOPSO	271.02	4907.98	1108.86	0.9504	0.9943	0.9705	109.87	113.28	111.71	0.487	−2.E-14	6.1995	0.02	11	11	27	35	2	2	1	0
E4	NSGA-III	267.98	6911.04	2771.29	0.9500	0.9950	0.9659	129.60	133.24	131.92	0.619	1.E−15	5.3687	0.00	30	30	49	60	8	8	9	9
	MOPSO	280.24	6911.04	1294.93	0.9500	0.9930	0.9733	129.94	133.12	131.74	0.381	−2.E−14	8.5333	0.2	10	10	29	36	2	2	1	1
F1	NSGA-III	1000.13	6911.04	3049.42	0.9500	0.9854	0.9620	96.66	98.79	98.03	0.500	−3.E−14	4.7774	0.00	24	24	28	50	9	9	10	10
	MOPSO	1025.34	5508.16	2429.60	0.9502	0.9852	0.9643	96.69	98.72	97.92	0.500	−8.E−15	1.8713	0.00	10	10	15	18	1	1	0	0
F2	NSGA-III	333.44	6911.04	2443.88	0.9500	0.9939	0.9647	95.58	98.94	97.80	0.500	−6.E−15	6.3368	0.00	26	26	49	51	9	9	8	8
	MOPSO	333.84	5089.26	1094.63	0.9508	0.9914	0.9724	95.66	98.63	97.45	0.500	−2.E−15	10.868	0.00	12	12	42	17	1	1	2	2
F3	NSGA-III	267.98	6911.04	2455.60	0.9500	0.9950	0.9650	95.67	99.42	98.17	0.500	−4.E−14	5.1552	0.00	25	25	47	50	8	8	8	9
	MOPSO	275.87	5684.99	1276.96	0.9500	0.9946	0.9737	96.12	99.15	98.12	0.500	−5.E−15	10.779	0.00	13	13	15	19	2	2	2	1
F4	NSGA-III	267.98	6911.04	2565.09	0.9500	0.9949	0.9652	96.96	100.84	99.52	0.521	−2.E−14	6.1172	0.00	25	25	40	45	7	7	9	10
	MOPSO	267.98	4830.38	1013.97	0.9512	0.9941	0.9751	96.99	100.49	99.24	0.479	−1.E−14	7.5280	0.03	15	15	19	18	3	3	1	0
G1	NSGA-III	267.98	6911.04	3138.41	0.9100	0.9950	0.9442	95.00	99.08	97.54	0.500	2.E−14	4.8658	0.00	30	30	49	62	6	6	7	8
	MOPSO	268.91	6221.26	1745.82	0.9133	0.9949	0.9610	95.41	98.98	97.45	0.500	3.E−14	2.9351	0.00	18	18	42	35	4	4	3	2
G2	NSGA-III	267.98	6911.04	2870.40	0.9300	0.9950	0.9540	95.02	99.08	97.65	0.503	−2.E−14	5.0952	0.00	28	28	50	52	7	7	7	9
	MOPSO	275.24	6911.04	1712.13	0.9303	0.9928	0.9619	95.66	98.86	97.49	0.498	−2.E−14	5.2373	0.00	16	16	43	45	3	3	3	1
G3	NSGA-III	267.98	3085.53	1180.05	0.9700	0.9950	0.9783	96.01	98.96	98.00	0.505	−4.E−15	2.1703	0.00	28	28	49	53	7	7	8	9
	MOPSO	267.98	2522.73	674.04	0.9703	0.9936	0.9823	96.04	98.84	97.81	0.495	3.E−15	6.9377	0.02	15	15	42	43	3	3	2	1
G4	NSGA-III	267.98	614.14	400.49	0.9900	0.9950	0.9918	97.80	98.82	98.43	0.513	7.E−16	0.2191	0.00	27	27	49	60	8	8	8	9
	MOPSO	267.98	608.00	353.00	0.9900	0.9950	0.9923	97.80	98.82	98.32	0.487	1.E−15	0.2241	0.04	12	12	40	36	2	2	2	1
H1	NSGA-III	267.98	872.79	481.97	0.9500	0.9772	0.9591	95.32	96.75	96.23	0.503	−1.E−15	0.5107	0.00	35	35	48	50	8	8	9	10
	MOPSO	267.98	861.83	416.59	0.9500	0.9759	0.9622	95.38	96.69	96.16	0.498	2.E−15	0.6650	0.05	17	17	36	30	2	2	1	0
H2	NSGA-III	267.98	2602.68	1012.89	0.9500	0.9892	0.9626	95.32	97.85	97.00	0.503	−1.E−15	2.2206	0.00	36	36	52	59	8	8	9	9
	MOPSO	267.98	1864.59	650.23	0.9501	0.9892	0.9680	95.35	97.72	96.86	0.498	2.E−15	2.3898	0.03	18	18	25	25	2	2	1	1
H3	NSGA-III	267.98	6911.04	3187.99	0.9500	0.9978	0.9695	96.03	100.17	98.55	0.500	4.E−15	4.7595	0.00	30	30	56	59	8	8	8	9
	MOPSO	269.67	6203.18	1415.66	0.9515	0.9967	0.9801	96.15	99.83	98.44	0.500	1.E−14	6.7312	0.00	15	15	26	29	2	2	2	1
H4	NSGA-III	267.98	6911.04	3731.58	0.9500	0.9990	0.9742	97.09	101.33	99.25	0.510	4.E−15	4.3407	0.00	32	32	58	60	9	9	10	10
	MOPSO	281.39	6911.04	1958.98	0.9530	0.9985	0.9848	97.24	101.15	99.30	0.490	−5.E−14	6.9700	0.04	16	16	31	30	1	1	0	0
I1	NSGA-III	267.98	6911.04	2989.02	0.9500	0.9950	0.9669	96.08	105.63	101.77	0.704	−2.E−14	4.7289	0.00	34	34	45	50	9	9	10	10
	MOPSO	268.08	4463.94	1059.85	0.9505	0.9926	0.9726	96.81	104.82	100.57	0.296	2.E−14	6.9326	0.31	20	20	26	28	1	1	0	0
I2	NSGA-III	267.98	6911.04	2741.91	0.9500	0.9950	0.9658	95.33	100.63	98.71	0.581	−6.E−15	4.6192	0.00	32	32	49	55	9	9	10	10
	MOPSO	267.98	6091.87	1186.49	0.9520	0.9946	0.9736	95.70	100.37	98.19	0.419	1.E−14	8.7703	0.18	25	25	25	24	1	1	0	0
I3	NSGA-III	267.98	6911.04	2538.67	0.9500	0.9943	0.9649	95.64	99.38	98.09	0.529	1.E−14	6.3299	0.00	35	35	48	49	8	8	10	10
	MOPSO	268.06	4088.37	860.71	0.9501	0.9943	0.9762	96.30	99.18	97.86	0.471	9.E−16	5.4213	0.05	22	22	25	26	2	2	0	0
I4	NSGA-III	267.98	6911.04	2699.09	0.9500	0.9950	0.9654	96.48	100.25	98.89	0.592	−3.E−15	4.1406	0.00	38	38	49	59	9	9	10	10
	MOPSO	269.31	5865.03	1084.73	0.9515	0.9942	0.9742	97.15	99.93	98.70	0.408	2.E−14	17.584	0.15	21	21	23	32	1	1	0	0
J1	NSGA-III	200.00	3142.51	1396.88	0.9500	0.9962	0.9674	94.94	98.96	97.52	0.500	2.E−15	2.4286	0.00	36	36	49	66	6	6	7	8
	MOPSO	211.74	3042.46	836.66	0.9502	0.9937	0.9750	95.51	98.88	97.48	0.500	−1.E−14	4.3484	0.00	17	17	40	35	4	4	3	2
J2	NSGA-III	200.21	4638.23	1824.65	0.9500	0.9962	0.9663	94.92	99.01	97.62	0.500	2.E−14	2.5431	0.00	38	38	50	52	7	7	7	9
	MOPSO	200.20	4136.57	1001.02	0.9527	0.9950	0.9738	95.31	98.84	97.49	0.500	−2.E−14	6.4774	0.00	19	19	43	45	3	3	3	1
J3	NSGA-III	370.40	7849.67	2733.28	0.9500	0.9934	0.9640	95.73	99.10	97.99	0.508	7.E−15	6.8648	0.00	40	40	49	53	7	7	8	9
	MOPSO	370.40	7279.81	1403.87	0.9516	0.9915	0.9713	96.17	98.88	97.76	0.492	4.E−14	19.591	0.07	20	20	42	43	3	3	2	1
J4	NSGA-III	516.74	7850.51	2855.49	0.9500	0.9905	0.9634	96.13	99.10	98.11	0.500	1.E−14	7.3638	0.00	38	38	48	62	8	8	8	10
	MOPSO	370.40	6058.36	1190.29	0.9507	0.9908	0.9723	96.12	98.99	97.64	0.500	2.E−14	10.469	0.00	18	18	35	31	2	2	2	0
Average	NSGA-III	301.44	6279.09	2472.55	0.9086	0.9543	0.9243	105.75	113.58	110.67	0.547	−4.E−15	4.7514	0.0003	28.5	28.5	43.3	50.8	7.3	7.3	8.3	8.8
	MOPSO	293.61	5303.75	1178.90	0.9096	0.9548	0.9330	106.34	113.10	109.96	0.453	−5.E−15	9.0440	0.0864	15.9	15.9	29.6	30.6	2.6	2.6	1.6	0.9

References

Asadzadeh, S., & Khoshalhan, F. (2009). Multiple-objective design of an X control chart with multiple assignable causes. *The International Journal of Advanced Manufacturing Technology, 43*(3-4), 312–322.

Banker, R. D. (1984). Estimating most productive scale size using data envelopment analysis. *European Journal of Operational Research, 17*(1), 35–44.

Castillo, E. D., Mackin, P., & Montgomery, D. C. (1996). Multiple-criteria optimal design of X control charts. *IIE transactions, 28*(6), 467–474.

Celano, G., & Fichera, S. (1999). Multiobjective economic design of an X control chart. *Computers & Industrial Engineering, 37*(1), 129–132.

Charnes, A., Cooper, W. W., & Rhodes, E. (1978). Measuring the efficiency of decision making units. *European Journal of Operational Research, 2*(6), 429–444.

Chen, Y. K., & Liao, H. C. (2004). Multi-criteria design of an X control chart. *Computers & Industrial Engineering, 46*(4), 877–891.

Chen, Y. S., & Yang, Y. M. (2002). Economic design of x-control charts with Weibull in-control times when there are multiple assignable causes. *International Journal of Production Economics, 77*(1), 17–23.

Chung, K. J. (1994). An algorithm for computing the economically optimal X-control chart for a process with multiple assignable causes. *European Journal of Operational Research, 72*(2), 350–363.

Coello, C. A. C. (1999). A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems, 1*(3), 269–308.

Coello, C. A. C., Pulido, G. T., & Lechuga, M. S. (2004). Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation, 8*(3), 256–279.

Cook, W. D., & Seiford, L. M. (2009). Data envelopment analysis (DEA)–Thirty years on. *European Journal of Operational Research, 192*(1), 1–17.

Cook, W. D., Tone, K., & Zhu, J. (2014). Data envelopment analysis: prior to choosing a model. *Omega, 44*, 1–4.

Das, I., & Dennis, J. E. (1998). Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization, 8*(3), 631–657.

Deb, K., & Jain, H. (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. *IEEE Transactions on Evolutionary Computation, 18*(4), 577–601.

Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. *Lecture Notes in Computer Science, 1917*, 849–858.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation, 6*(2), 182–197.

Duncan, A. J. (1956). The economic design of X charts used to maintain current control of a process. *Journal of the American Statistical Association, 51*(274), 228–242.

Duncan, A. J. (1971). The economic design of-charts when there is a multiplicity of assignable causes. *Journal of the American Statistical Association, 66*(333), 107–121.

Durillo, J. J., García-Nieto, J., Nebro, A. J., Coello, C. A. C., Luna, F., & Alba, E. (2009). Multi-objective particle swarm optimizers: an experimental comparison. *Evolutionary multi-criterion optimization* (pp. 495–509). Berlin Heidelberg: Springer.

Faraz, A., & Saniga, E. (2013). Multiobjective genetic algorithm approach to the economic statistical design of control charts with an application to X-bar and S2 Charts. *Quality and Reliability Engineering International, 29*(3), 407–415.

Fonseca, C. M., & Fleming, P. J. (1995). An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation, 3*(1), 1–16.

Goh, C. K., & Tan, K. C. (2009). A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization. *IEEE Transactions on Evolutionary Computation, 13*(1), 103–127.

Ho, C. (1994). Economic design of control charts: A literature review for 1981-1991. *Journal of Quality Technology, 26*(1), 39–53.

Hwang, C. L., & Masud, A. S. M. (1979). *Multiple objective decision making—methods and applications*. Springer.

Jain, H., & Deb, K. (2014). An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: handling constraints and extending to an adaptive approach. *IEEE Transactions on Evolutionary Computation, 18*(4), 602–622.

Jones, D. F., Mirrazavi, S. K., & Tamiz, M. (2002). Multi-objective meta-heuristics: an overview of the current state-of-the-art. *European Journal of Operational Research, 137*(1), 1–9.

Kennedy, J., Kennedy, J. F., Eberhart, R. C., & Shi, Y. (2001). *Swarm intelligence*. Morgan Kaufmann.

Li, Z., Kapur, K. C., & Chen, T. (2009). A new approach for multicriteria design of an X control chart. In *Proceedings of the 8th international conference on reliability, maintainability and safety (ICRMS)* (pp. 1364–1367).

Lorenzen, T. J., & Vance, L. C. (1986). The economic design of control charts: a unified approach. *Technometrics, 28*(1), 3–10.

McMullen, P. R. (2001). An ant colony optimization approach to addressing a JIT sequencing problem with multiple objectives. *Artificial Intelligence in Engineering, 15*(3), 309–317.

Michalewicz, Z. (1996). *Genetic algorithms+ data structures = evolution programs*. Springer Science & Business Media.

Mobin, M., Li, Z., & Massahi Khoraskani, M. (2015). Multi-objective X-bar control chart design by integrating NSGA-II and data envelopment analysis. In *Proceedings of the 2015 industrial and systems engineering research conference (ISERC)* (pp. 1–10).

Montgomery, D. C. (1980). The economic design of control charts: a review and literature survey. *Journal of Quality Technology, 12*(2), 75–87.

Moore, J., & Chapman, R. (1999). *Application of particle swarm to multiobjective optimization*. Department of Computer Science and Software Engineering, Auburn University.

Naka, S., Genji, T., Yura, T., & Fukuyama, Y. (2001). Practical distribution state estimation using hybrid particle swarm optimization. In *Power engineering society winter meeting (IEEE): 2* (pp. 815–820).

Pearn, W. L., & Chen, K. S. (1997). Capability indices for non-normal distributions with an application in electrolytic capacitor manufacturing. *Microelectronics Reliability, 37*(12), 1853–1858.

Safaei, A. S., Kazemzadeh, R. B., & Niaki, S. T. A. (2012). Multi-objective economic statistical design of X-bar control chart considering Taguchi loss function. *The International Journal of Advanced Manufacturing Technology, 59*(9-12), 1091–1101.

Salmon, C., Mobin, M., & Roshani, A. (2015). TOPSIS as a Method to Populate Risk Matrix Axes. In *Proceedings of the industrial and systems engineering research conference (ISERC)* (pp. 1–10).

Saniga, E. M. (1977). Joint economically optimal design of X and R control charts. *Management Science, 24*(4), 420–431.

Saniga, E. M. (1989). Economic statistical control-chart designs with an application to and R charts. *Technometrics, 31*(3), 313–320.

Scott, J. R. (1995). *Fault tolerant design using single and multi-criteria genetic algorithms* Master's thesis,. Department of Aeronautics and Astronautics, Massachusetts Institute of Technology.

Shi, Y., & Eberhart, R. C. (1999). Empirical study of particle swarm optimization. In *Proceedings of the 1999 congress on evolutionary computation (CEC)* (pp. 1945–1950).

Sierra, M. R., & Coello, C. A. C. (2005). Evolutionary Multi-Criterion Optimization, Volume 3410 of the series Lecture Notes in Computer Science. In C. A. C. Coello, A. H. Aguirre, & E. Zitzler (Eds.), *Proceedings of the Third International Conference, EMO 2005, Guanajuato, Mexico, March 9-11*.

Srinivas, N., & Deb, K. (1994). Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation, 2*(3), 221–248.

Woodall, W. H. (1986). Weaknesses of the economic design of control charts. *Technometrics, 28*(4), 408–409.

Yang, W. A., Guo, Y., & Liao, W. (2012). Economic and statistical design of and S control charts using an improved multi-objective particle swarm optimisation algorithm. *International Journal of Production Research, 50*(1), 97–117.

Yu, F. J., & Hou, J. L. (2006). Optimization of design parameters for control charts with multiple assignable causes. *Journal of Applied Statistics, 33*(3), 279–290.

Yuan, Y., Xu, H., Wang, B., & Yao, X. (2015). A new dominance relation based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation, 99*, 1.

Zhou, A., Qu, B. Y., Li, H., Zhao, S. Z., Suganthan, P. N., & Zhang, Q. (2011). Multiobjective evolutionary algorithms: a survey of the state of the art. *Swarm and Evolutionary Computation, 1*(1), 32–49.

Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation, 3*(4), 257–271.