Measurement 114 (2018) 417–427

Contents lists available at ScienceDirect

Measurement

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

A hybrid desirability function approach for tuning parameters in evolutionary optimization algorithms

Mohammadsadegh Mobin[a,*], Seyed Mohsen Mousavi[b], Mohammad Komaki[c], Madjid Tavana[d,e]

[a] Industrial Engineering and Engineering Management Department, Western New England University, Springfield, USA
[b] Department of Mechanical Engineering, University of Malaya, Kuala Lumpur, Malaysia
[c] Electrical Engineering and Computer Science Department, Case Western Reserve University, Cleveland, USA
[d] Business Systems and Analytics Department, Distinguished Chair of Business Analytics, La Salle University, Philadelphia, USA
[e] Business Information Systems Department, Faculty of Business Administration and Economics, University of Paderborn, Paderborn, Germany

ARTICLE INFO

Keywords:
Parameter tuning
Evolutionary algorithms
Desirability function
Multi-objective particle swarm optimization
Fast non-dominated sorting genetic algorithm
Multi-objective single machine scheduling

ABSTRACT

Evolutionary algorithms are optimization methods commonly used to solve engineering and business optimization problems. The parameters in evolutionary algorithm must be perfectly tuned in a way that the optimization algorithm solves the optimization problems efficiently and effectively. Several parameter tuning approaches with a single performance metric have been proposed in the literature. However, simultaneous consideration of multiple performance metrics could provide the optimal setting for the parameters in the evolutionary algorithm. In this research, a new hybrid parameter tuning approach is proposed to simultaneously optimize the performance metrics of the evolutionary optimization algorithm while it is used in solving an optimization problem. The proposed hybrid approach provides the optimal value of parameters of the evolutionary optimization algorithm. The proposed approach is the first parameter tuning approach in the evolutionary optimization algorithm which simultaneously optimizes all performance metrics of the evolutionary optimization algorithm. To do this, a full factorial design of experiment is used to find the significant parameters of the evolutionary optimization algorithm, as well as an approximate equation for each performance metric. The individual and composite desirability function approaches are then proposed to provide the optimal setting for the parameters of the evolutionary optimization algorithm. For the first time, we use the desirability function approach to find an optimal level for the parameters in the evolutionary optimization algorithm. To show the real application of the proposed parameter tuning approach, we consider two multi-objective evolutionary algorithms, i.e., a multi-objective particle swarm optimization algorithm (MOPSO) and a fast non-dominated sorting genetic algorithm (NSGA-III) and solve a single machine scheduling problem. We demonstrate the applicability and efficiency of the proposed hybrid approach in providing the optimal values of all parameters of the evolutionary optimization algorithms to optimize their performance in solving an optimization problem.

1. Introduction

One of the challenging, yet not appropriately investigated, questions in developing and applying an evolutionary optimization algorithm is to find the optimal setting of the parameters of the algorithms. This process, known as parameter tuning, has usually been considered as a difficult task, whether it be in developing a new evolutionary algorithm or a new application.

Evolutionary algorithms or meta-heuristic algorithms, which mimic the natural processes in the world, are based on a general framework that can be applied to all optimization problems; while other algorithms, such as some heuristic algorithms, are only applicable to some specific kinds of the problems. Each evolutionary algorithm starts with a group of initial solutions (or an individual solution) and iterates several operations on the solutions until a stopping condition is met. Some of the well-known meta-heuristic algorithms are the Genetic Algorithm (GA) [17], Simulated Annealing (SA) [30], and Particle Swarm Optimization (PSO) [14]. Each of these evolutionary algorithms is motivated by the behavior of different phenomena and has its own parameters which need to be tuned to achieve the optimal performance in solving an optimization problem. For instance, the Genetic Algorithm and Particle Swarm Optimization are population-based algorithms; therefore, one needs to obtain the optimal value of the population size, while the algorithm is used to solve the required optimization problem.

* Corresponding author at: Industrial and Engineering Management Department, Western New England University, Springfield, MA 01119, United States.
E-mail addresses: mm337076@wne.edu (M. Mobin), mousavi@siswa.um.edu.my (S.M. Mousavi), komakighorban@gmail.com (M. Komaki), tavana@lasalle.edu (M. Tavana).
URL: http://tavana.us/ (M. Tavana).

http://dx.doi.org/10.1016/j.measurement.2017.10.009
Received 21 June 2017; Received in revised form 3 August 2017; Accepted 3 October 2017
Available online 05 October 2017
0263-2241/

The performance of the evolutionary optimization algorithm depends to a great extent on the values of its parameters. Also, the parameter settings suitable for one optimization problem may not be suitable for another problem. Furthermore, the evolutionary algorithms are not easily applicable to many real-world engineering optimization problems because the functional forms of the objective functions are unknown or hard to estimate, e.g., laser-based additive manufacturing process optimization [1]. Therefore, choosing an appropriate parameter tuning of an algorithm should be taken into account for each specific optimization problem.

In the literature, there are a variety of approaches that have been used to tune the parameters of different evolutionary algorithms for single or multiple objective problems. Akay and Karaboga [2] investigated the performance of the artificial bee colony algorithm by analyzing the effect of the control parameters. Crawford et al. [11] applied a particle swarm optimization to adjust the parameters of a choice function based hyper-heuristic. Smit and Eiben [49] applied the relevance estimation and value calibration method, called REVAC, to find optimal parameter values of an evolutionary algorithm. Iwasaki et al. [25] proposed a dynamic parameter tuning method for the particle swarm optimization algorithm. The feasibility of their approach was investigation for a particle swarm optimization algorithm and was verified using numerical simulations using some typical global optimization problems. Vafadarnikjoo et al. [56] applied a full factorial design of experiments to tune the parameters of the artificial bee colony algorithm, while it was used to solve a switch location problem in a cellular mobile network. Yu and Seif [59] applied a design of experiment to tune the parameters of a genetic algorithm for a maintenance flow shop scheduling problem. Tavana et al. [53,54] proposed an effective artificial immune algorithm AIA) to classify ergonomic products with multi-criteria anthropometric measurements and tuned the AIA parameters with a full factorial experimental design approach. To reach the precise calibration of various operators and parameters of the AIA, a comprehensive comparison by using a full factorial design of experiment was performed. Kayvanfar et al. [29] conducted full factorial experiments so as to tune the significant parameters of the Intelligent Water Drops (IWD) algorithm for the identical parallel machine scheduling problem. Teymourian et al. [55] carried out full factorial experiments to tune the parameters of the IWD algorithm for the agile manufacturing system. More recently, Hassani and Jafarian [22] used a full factorial design of experiment to tune the parameters of GA and PSO to optimize the breast cancer diagnosis accuracy problem. A comprehensive comparison of parameter tuning methods for evolutionary algorithms were presented by [48,15].

Most of the tuning approaches applied in the literature have relied heavily on the methods motivated by previous studies. For example, Rashidi et al. [44] used a simulated annealing algorithm to solve the effect of placing sidewalks and crosswalks in a transportation network, using parameter values from other applications. Heydari et al. [23] considered the most commonly used parameter values for a simulated annealing algorithm to solve a resource allocation problem. To appropriately tune the parameters of an Ant Colony algorithm, Reihaneh and Karapetyan [45] and Salari et al. [46] tested a range of values used in the literature to obtain the parameters resulting in the best performance. Hajizadeh et al. [19] proposed a particle swarm optimization PSO) to tune the parameter in a neural network algorithm to forecast Euro/Dollar exchange rate volatility. Jafari-Marandi et al. [26] individually tuned the significant parameters of GA for each level by tracking the behavior of the algorithm for different ranges of these parameters. However, the parameter tuning process should be applied for each specific algorithm in solving each problem. Therefore, tuning parameters is very critical to obtain the optimal performance of an evolutionary algorithm.

The Taguchi method [52], which is based on design of experiment concept, has been used in the literature to tune the parameters of the evolutionary optimization algorithm. There are a variety of works in the literature that used the Taguchi approach to set the parameters of the different algorithms in single-objective problems [37,42]. Kayvanfar and Zandieh [28] used the Taguchi approach in order to calibrate the parameters of the applied Imperialism Competitive Algorithm for the economic lot scheduling problem. Gohari and Salmasi [16] and Naderi et al. [40] utilized the Taguchi approach to tune the parameters of different PSO-based hybrid optimization algorithms in the flexible flow line problem. However, the Taguchi approach is applicable in handling the nuisance factors in an experiment, while the parameters of the evolutionary algorithm are the main factors. Therefore, the Taguchi approach is not an effective tool to tune the parameters of an evolutionary optimization algorithm.

From the reviewed literature, it can be seen that almost all parameter tuning approaches only consider one performance metric of the algorithms, while finding the optimal setting of the parameters. However, there are multiple performance metrics that can be used to evaluate the performance of an evolutionary algorithm. These performance metrics include the mean ideal distance, spacing, and spread that need to be taken into consideration simultaneously when tuning the parameters of an algorithm. This necessitates application of multi-response optimization approaches in the parameter tuning of the evolutionary optimization algorithm. In this research, the application of a composite desirability function approach, which is a multi-response optimization approach, is proposed, which optimizes all performance metrics of the evolutionary optimization algorithm, while providing the optimal setting of the parameters of the algorithm. In this regard, obtaining an estimated regression model for each performance metric is required. In the proposed approach, a full factorial design of experiment is applied to identify the significant parameters of the algorithms, which plays a role in the approximated regression function.

The most significant contribution of the proposed hybrid parameter tuning approach is that it simultaneously considers all performance metrics (response variables) of the evolutionary optimization algorithm in tuning the parameters (factors) of the evolutionary optimization algorithm. The existing parameter tuning approaches in the optimization literature do not consider all performance metrics while tuning the parameters, but rather, provide the optimal setting of parameters based on the individual consideration of each performance metric. Therefore, instead of providing a general optimal parameter setting of the evolutionary optimization algorithm which can be obtained by the proposed approach in this paper, the traditional parameter tuning approaches provide different setting of values, while each setting of values only satisfies one performance metric of the evolutionary optimization algorithm.

The rest of this paper is organized as follows: Section 2 is devoted to develop the methodology to find the optimal parameters of evolutionary algorithms. Section 3 presents a case study problem. The description of the investigated evolutionary algorithms, i.e., multi-objective particle swarm optimization algorithm (MOPSO) and the fast non-dominated sorting genetic algorithm (NSGA-III), to solve the case study problem are provided in Section 4. The parameters of the MOPSO and NSGA-III algorithms and the performance metrics to evaluate the performance of the algorithms are provided in Section 5. The application of the proposed methodology in the algorithms parameters tuning is illustrated in a case study in Section 6. Finally, Section 7 presents the conclusion and future research directions.

2. The proposed methodology to tune the parameters of the evolutionary optimization algorithms

In order to obtain the optimal setting of the parameters for the evolutionary algorithms, some metrics are measured for evaluating the performance of these algorithms in solving the optimization problem. Each parameter can be considered as a factor, each with different predefined levels. The performance metrics are considered as the response variables. Considering all the combinations of the factor levels, a full factorial design of experiment can be conducted to investigate the effect of each factor on the response variables, obtain the significant factors, and approximate the regression model for each response variable. The details of the full factorial design description and applications can be found in [58,36]. Then, considering each regression model as an objective function, the desirability function approach is applied to optimize all the response variables simultaneously, and obtain the optimal values for all factors.

The desirability function approach (DFA) was first introduced by Harrington [20] and extended later by Derringer and Suich [13]. This method is a search-based optimization method which optimizes multiple response variables, individually and simultaneously, to find the optimum input variable settings. The desirability function approach has been used in several applications, mostly to optimize a manufacturing process. For example, this approach has been used to investigate and optimize the mechanical surface treatment processes [6,31,36]. Other applications of the desirability function approach in optimizing various manufacturing processes can be found in Ramanujam et al. [43], Boulet et al. [9], and Balamugundan et al. [8]. A review of applications of robust design methods for multiple responses, including the desirability function method, is presented by Murphy et al. [39].

An analysis of the multiple response optimization process includes creating a mathematical model, known as a regression model, for each response variable, and then obtaining a set of factors to optimize all the responses. To solve the multi-response optimization problems, a technique for integrating multiple responses into a dimensionless function, called the overall desirability function (D), is applied. The approach is to convert each response (y_i) into a dimensionless function, known as the individual desirability function (d_i), that can be between zero and one. If the response y_i is at its target the most desirable case is obtained ($d_i = 1$), otherwise, $d_i = 0$ (the least desirable case). The desirability function approach assumes that there is a positive number, w, known as the weight factor [36]. To simplify our investigation, the weights for the response variables are considered equal to one.

In this study, the individual desirability functions are calculated based on the type of the optimization functions, i.e. maximization or minimization using Eqs. (1)–(3). If the target (T_i) for the response y_i is a maximum value, the desirability is based on Eq. (1). If the target is a minimization one, the desirability is based on Eq. (2). Furthermore, if the target is located between the lower (L_i) and upper (U_i) limits, the desirability is obtained based on Eq. (3).

$$d_i = \begin{cases} 0 & y_i < L_i \\ \left(\frac{y_i - L_i}{T_i - L_i}\right)^w & L_i \leqslant y_i \leqslant T_i \\ 1 & y_i > T_i \end{cases} \tag{1}$$

$$d_i = \begin{cases} 1 & y_i < L_i \\ \left(\frac{T_i - y_i}{T_i - L_i}\right)^w & L_i \leqslant y_i \leqslant T_i \\ 0 & y_i > T_i \end{cases} \tag{2}$$

$$d_i = \begin{cases} 0 & y_i < L_i \\ \left(\frac{y_i - L_i}{T_i - L_i}\right)^w & L_i \leqslant y_i \leqslant T_i \\ \left(\frac{U_i - y_i}{U_i - L_i}\right)^w & T_i \leqslant y_i \leqslant U_i \\ 0 & y_i > U_i \end{cases} \tag{3}$$

Next, the individual desirability functions are integrated as overall (composite or aggregated) desirability (D), which can be between 0 and 1. It is defined as the weighted geometric mean of all the previously defined desirability functions, calculated by Eq. (4), where w_i is a comparative scale for weighing each of the resulting d_i assigned to the i th response, and n is the number of responses. The optimal values of the parameters are determined to maximize overall desirability (D), by applying a reduced gradient algorithm with multiple starting points.

$$D = \left(d_1^{w_1} \times d_2^{w_2} \times d_3^{w_3} \times \ldots \times d_n^{w_n}\right)^{\frac{1}{(w_1+w_2+w_3+\ldots+w_n)}} = \prod_{i-1}^{n} d_i^{w_i \frac{1}{\sum_{i=1}^{n} w_i}} \tag{4}$$

More details about the desirability function approach and its applications are presented in [36].

3. Case study

In this section, a three-objective single machine scheduling problem is solved to minimize the makespan, total completion times, and total tardiness times. In this problem, there is a single machine and N jobs available at time 0 such that all jobs should be processed by one machine at a time. Each job $j \in \{1,2,\ldots,N\}$ has processing time p_j, due date d_j and importance weight w_j. Also, there is a sequence dependent set up time between jobs. Assuming p_i is a sequence of jobs, where $\pi(1)$ represents the job in the first position, then the objectives of sequence π can be computed as follows:

$$C(\pi(i)) = \sum_{k=1}^{i} (p_{\pi(k)} + S_{\pi(k),\pi(k-1)}) \quad \text{for} \quad i = 1,2,\ldots,N \tag{5}$$

$$T(\pi(i)) = max\{C(\pi(i)) - d(\pi(i)), 0\} \quad \text{for} \quad i = 1,2,\ldots,N \tag{6}$$

Eqs. (5) and (6) compute the completion time and tardiness of job in position i of the sequence π, respectively. Note that $S_{\pi(k),\pi(k-1)}$ represents the set up time required to process job $\pi(k)$ immediately after job $\pi(k-1)$. Then, the makespan of the sequence π is:

$$C_{max} = C(\pi(N)) \tag{7}$$

The total weighted completion time (TWC) of the sequence π is as follows:

$$TWC = \sum_{i=1}^{N} w_{\pi(i)} C(\pi(i)) \tag{8}$$

The total weighted tardiness (TWT) of the sequence is formulated as follows:

$$TWT = \sum_{i=1}^{N} w_{\pi(i)} T(\pi(i)) \tag{9}$$

Using the three-field problem classification $\alpha|\beta|\gamma$ of Graham et al. [18], the addressed problem can be presented as $1|d,S|(Cmax,TWT,TWC)$ and has been shown to be strongly NP-hard [32].

4. Evolutionary optimization algorithms

In this study, two Multi-Objective Evolutionary Algorithms (MOEAs) are used to solve the problem described in the previous section. MOPSO and NSGA-III are the most common algorithms in the literature used to solve the problems in the field [24,21]. In this research, a MOPSO [10] and NSGA-III [53,54] are applied to find the Pareto optimal solutions of the problem. The algorithms are described in the following subsections.

4.1. Multi-objective particle swarm optimization (MOPSO) algorithm

MOPSO is a population-based meta-heuristic algorithm inspired from the social behavior of birds which has been used frequently in the recent literature [4,5,7,24,33,35,38,51,57]. Each bird (or particle) represents a solution of the problem. The algorithm starts with an initial population of particles which is randomly generated. The number of particles (solutions) N_{pop} is one of the MOPSO parameters that needs to be set. MOPSO is an iterative algorithm, that is, it iterates It_{max} times by performing the same operations at each iteration. The location of each particle b at each iteration t of the algorithm is represented by s_b^t and its velocity is represented by v_b^t. In this paper, the initial position and velocity of each particle $b = 1,2,...,N_{pop}$ is generated randomly and then the largest rule is applied to the s_b^t to find the sequence of jobs.

Each solution is evaluated based on its objective functions and a set of non-dominated solutions is generated. This set is called Repository of Particles (REP) [10]. The REP has two components; an archive controller and grid. The archive controller determines whether a solution can be added to the REP or not; a solution can be added to the REP if it is a non-dominated solution compared to the solutions in the REP. The grid controls the distribution of the Pareto frontier and it is a set of connected hypercubes, where each hypercube has some non-dominated solutions.

In each iteration of the algorithm, each particle moves toward the best location identified so far by itself or toward the best position found so far by the swarm. The best position found by the particle itself up to iteration t is represented by $pBest_b^t$ and the best position found by the swarm is represented by $gBest^t$. Usually, $gBest^t$ is randomly selected by using a roulette-wheel selection mechanism from a hypercube in the REP with the fewest members.

The velocity of each particle helps it to move toward the $pBest_b^t$ and $gBest^t$. At each iteration t of the algorithm, the velocity of the particle b is updated as follows:

$$v_b^t = w_t * v_b^{t-1} + C_1 * r_1 * (pBest_b^t - s_b^{t-1}) + C_2 * r_2 * (gBest^t - s_b^{t-1}) \tag{10}$$

where r_1 and r_2 are random numbers between 0 and 1, and C_1 and C_2 are constants to control the effects of $pBest_b^t$ and $gBest^t$, respectively. Note that w_t in Eq. (11), called the inertia weight of the particle b, controls the effect of the velocity of the particles at the previous iteration. Shi and Eberhart [47] and Naka et al. [41] suggested using a linearly decreasing weight as presented below:

$$w_t = w_{max} - \frac{w_{max} - w_{min}}{It_{max}} * t \tag{11}$$

where w_{min} and w_{max} represent the lower and upper bounds of w_t. The suggested values for these parameters are $w_{min} = 0.4$ and $w_{max} = 0.9$. After updating the velocity of the particle, the new position of the particle is shown as follows:

$$s_b^t = s_b^{t-1} + v_b^t \tag{12}$$

4.2. Fast non-dominated sorting genetic algorithm (NISGA-III)

Since the development of the genetic algorithm, there has been a growing interest in applying it in various applications and improving its performance by proposing new versions. For example, the Non Dominated Sorting Genetic Algorithm NSGA), developed by Srinivas and Deb [50], uses a non-dominated sorting procedure and applies a ranking method that emphasizes the good solutions, and tries to maintain them in the population. An extended version of NSGA, called NSGA-II, developed by Deb et al. [12], utilizes a fast non-dominated sorting genetic algorithm and 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 Jain and Deb [27], which is more efficient to solve problems with more than two objectives. The significant innovation of NSGA-III is the utilization of reference points which could be a set of predefined points, or one that are generated systematically. The procedure of NSGA-III algorithm is presented as follows. Note that all the steps of NSGA-II and NSGA-III are almost identical except for the selection mechanism, where the first one is based on the crowding distance and the latter one is based on reference points. These reference points are generated using systematic methods, such that they are on the surface of the normal hyperplane covering the entire area. Each solution is then assigned to each reference point, and afterwards, solutions are selected in a way that each solution has a representative in the solution of the next generation.

In the NSGA-III algorithm, first a population including N_{Pop} chromosomes called P_0 is generated randomly. Then, the offspring Q_t is generated using crossover and mutation operators. In this study, an arithmetic crossover operator introduced by Michalewicz and Hartley [34] is applied where each time two chromosomes s_i and s_r are chosen from the population randomly to create two offspring q_i and q_r, as follows (Eqs. (13) and (14)):

$$q_i = (\beta)s_i + (1-\beta)s_r \tag{13}$$

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

In the aforementioned equations, β is an uniform random number in the range [0,1]. The generated offspring q_i and q_r are saved to Q_t. After performing the crossover operator, a mutation operator is applied on the new offspring where a Gaussian mutation operator is used.

It should be mentioned that there are no constraints in the case study problem. However, in the case of constraints, the method proposed by Jain and Deb [27] can be used, where the constraints are normalized using Eqs. (15)–(17). Therefore, for each chromososme s_i,

Table 1
MOPSO parameters (factors) and their levels.

Factors	Title	Low level	High level
A	N_{pop}: Number of population	100	200
B	It_{max}: Max. iteration of algorithm	100	200
C	C_1: Control parameter of $pBest_b^t$	1	2
D	C_2: Control parameter of $gBest_b^t$	1	2

Table 2
NSGA-III parameters (factors) and their levels.

Factors	Title	Low level	High level
E	N_{pop}: Number of population	100	200
F	It_{max}: Max. iteration of algorithm	100	200
G	*Pro. Crs*: Probability of crossover	0.7	0.8
H	*Pro. Mut*: Probability of mutations	0.3	0.4

Table 3
Response variables (performance metrics) for the MOEA parameter tuning (NSGA-III and MOPSO).

	R1: MID	R2: *sp*	R3: Δ
Goal	Minimization	Minimization	Minimization
Target	0	0	0

Table 4
Factor combinations and response variables in MOPSO.

Run	Factors (algorithm parameters)				Response variables		
	A	B	C	D	R1	R2	R3
1	100	100	1	1	929.6198	130.4420	0.427543
2	200	100	1	1	947.9585	195.2630	0.349748
3	100	200	1	1	870.1583	129.1685	0.408271
4	100	100	2	1	872.9819	0.103935	0.406325
5	100	100	1	2	1042.207	278.8177	0.326788
6	200	200	1	1	724.0536	172.4374	0.478954
7	200	100	2	1	880.3909	156.6711	0.495372
8	200	100	1	2	863.4142	307.2580	0.523566
9	100	200	1	2	977.3852	332.8447	0.504895
10	100	200	2	1	1039.147	70.12113	0.354510
11	100	100	2	2	1023.136	235.6880	0.284067
12	200	200	2	1	802.6965	145.6140	0.413133
13	200	200	1	2	855.5707	303.4513	0.455522
14	200	100	2	2	958.9864	300.2676	0.371968
15	100	200	2	2	1015.849	184.0285	0.341144
16	200	200	2	2	893.6353	322.6226	0.378348

Table 5
Factor combinations and response variables in NSGA-III.

Run	Factors (algorithm parameters)				Response variables		
	E	F	G	H	R1	R2	R3
1	100	100	0.7	0.3	968.317	215.545	0.500000
2	100	100	0.7	0.4	879.844	278.356	0.500000
3	100	100	0.8	0.3	873.532	298.365	0.500000
4	100	100	0.8	0.4	877.981	416.903	0.726805
5	100	200	0.7	0.3	925.893	206.690	0.272727
6	100	200	0.7	0.4	860.786	209.850	0.500000
7	100	200	0.8	0.3	877.290	116.500	0.530456
8	100	200	0.8	0.4	883.616	191.500	0.578191
9	200	100	0.7	0.3	916.775	314.464	0.464104
10	200	100	0.7	0.4	823.610	212.342	0.678258
11	200	100	0.8	0.3	902.348	327.450	0.514557
12	200	100	0.8	0.4	873.159	364.125	0.642983
13	200	200	0.7	0.3	562.994	298.659	0.500000
14	200	200	0.7	0.4	830.137	273.200	0.634917
15	200	200	0.8	0.3	849.351	103.000	0.485751
16	200	200	0.8	0.4	790.043	179.009	0.682927

the constraint violation value ($CV(s_i)$) is computed by Eq. (15), where $\langle x \rangle$ is $-x$ if $x < 0$, and 0 otherwise.

$$g_1(s_i) = \frac{p(s_i)}{p_L} - 1 \geqslant 0, \quad \forall s_i \tag{15}$$

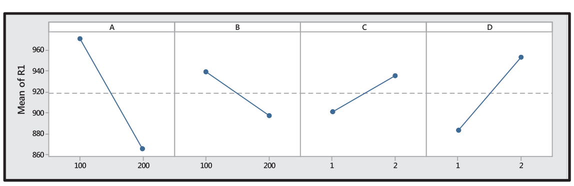
$$g_2(s_i) = -\alpha(s_i)/\alpha_U - 1 \geqslant 0, \quad \forall s_i \tag{16}$$

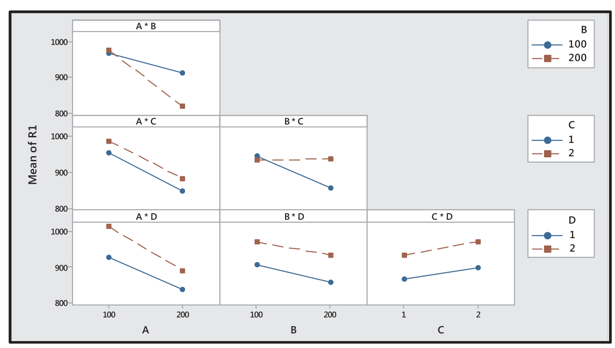
Table 6
FFT results considering response variable individually in tuning the parameters of MOPSO.

Responses	Factors Recommended level				Significant factors and interactions
	A	B	C	D	
R1	200	200	1	1	A, B, C, D, A * B
R2	100	–	2	1	A, C, D, A * C
R3	100	100	2	2	A, B, C, D

$$CV(s_i) = \langle g_1(s_i) \rangle + \langle g_2(s_i) \rangle \tag{17}$$

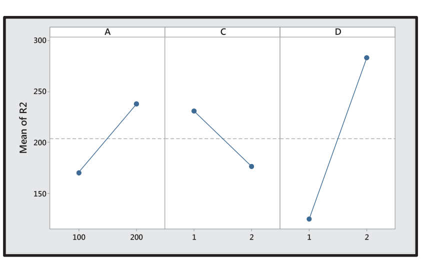
In order to generate population R_t with size of $2*N_{Pop}$, the parent population P_t and the offspring Q_t are combined where the fast non-dominated sorting is utilized based on the Pareto concept to assign R_t to the different non-dominance fronts, i.e., F_1,F_2, and so on. In the next step, the population P_{t+1} is generated based on the fronts F_1,F_2, ... The chromosomes in the higher non-dominance fronts are assigned to S_t,while it reaches to the size of N_{Pop} or it exceeds N_{Pop} for the first time at the non-dominance level l. Chromosomes (individuals) in the fronts higher than l are simply put away, and $S_t \backslash F_l$ are chosen as the next generation P_{t+1}. While the size of the next generation P_{t+1} is equal to N_{Pop}, the algorithm continues with the next iteration by generating new children (while the stopping criterion of the algorithm is not met), otherwise, the rest of $N_{Pop}-|P_{t+1}|$ chromosomes are selected from F_l according to the reference points. While the problem objectives might

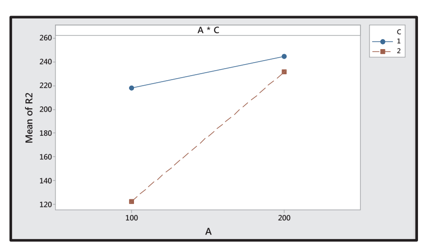




1.1 (a): The main effect plots for R1 1.1 (b): The interaction plots for R1

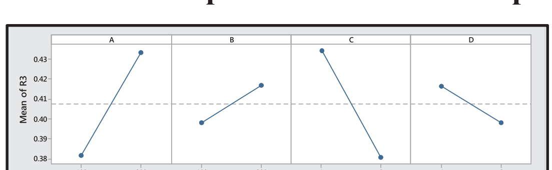
The main and interaction plots for the first response variable (MOPSO)





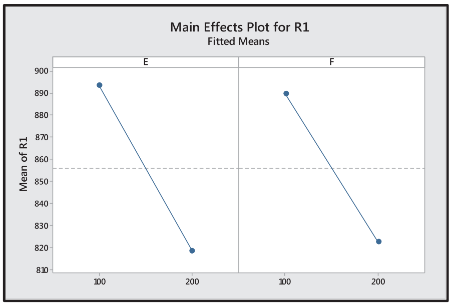
1.2(a): The main effect plots for R2 1.2(b): The interaction plots for R2

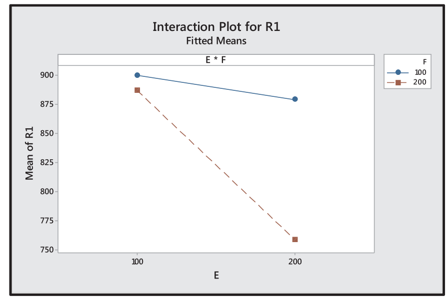
The main and interaction plots for the second response variable (MOPSO)



The main effect plot for the third response variable (MOPSO)

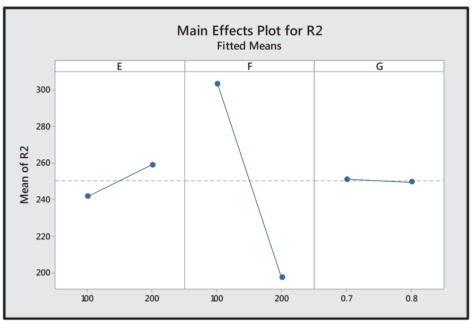
Fig. 1. The results of applying the full factorial design of experiment in investigating significant factors in the MOPSO.

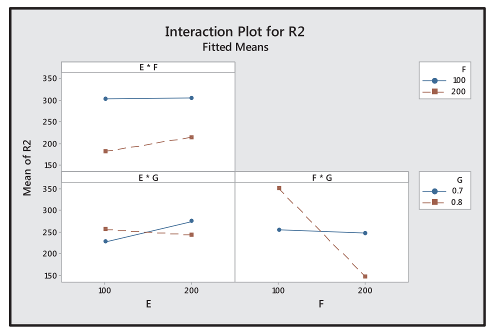




2.1(a): The main effect plots for R1 **2.1(b): The interaction plots for R1**

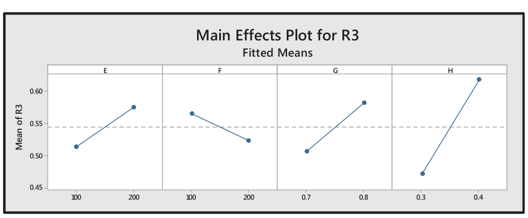
The main and interaction plots for the first response variable (NSGA-III)

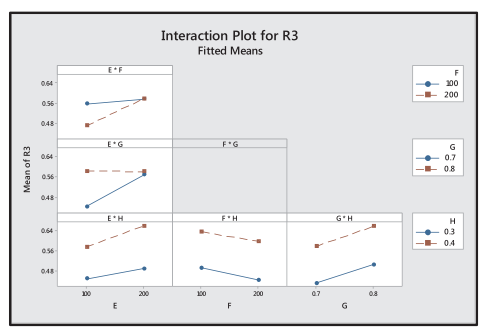




2.2(a): The main effect plots for R2 **2.2(b): The interaction plots for R2**

The main and interaction plots for the second response variable (NSGA-III)





2.3(a): The main effect plots for R2 **2.3(b): The interaction plots for R2**

The main and interaction plots for the third response variable (NSGA-III)

Fig. 2. The results of full factorial design of experiment in investigating significant factors in the NSGA-III.

Table 7
FFT results considering response variable individually in tuning the parameters of NSGA-III.

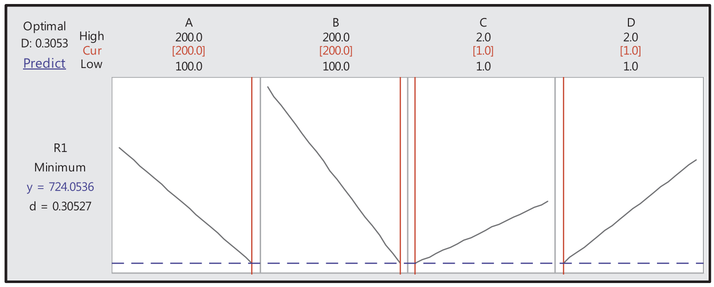
Responses	Factors Recommended level				Significant factors and interactions
	E	F	G	H	
R1	200	200	–	–	E, F, E * F
R2	100	200	0.8	–	E, F, G, E * F, E * G, F * G
R3	100	200	0.7	0.3	E * F, E * G, E * H

be different from each other in terms of type, the normalizing process and the reference points are created in the normalized range. Thus, each chromosome (solution) is allocated to a reference point. Then, the solutions of $S_t \setminus F_l$ and F_l are assigned to the nearest reference point and the rest of $N_{Pop} - |P_{t+1}|$ solutions in F_l are selected, so that their reference point does not have any relevant solution in $S_t \setminus F_l$. The readers who are interested in the normalization process of the objectives and choosing the rest of $N_{Pop} - |P_{t+1}|$ solutions in F_l are referred to [53,54].

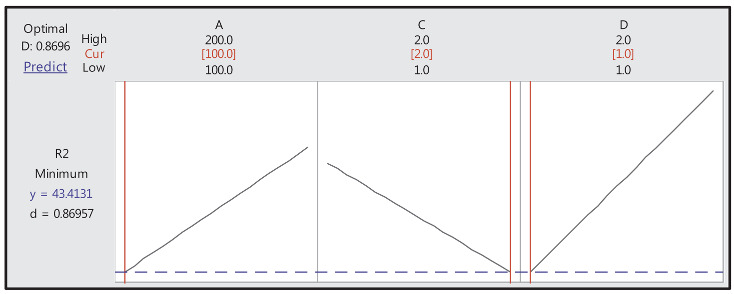
5. Parameters and performance metrics of evolutionary optimization algorithms

5.1. Parameters to be tuned in evolutionary optimization algorithms

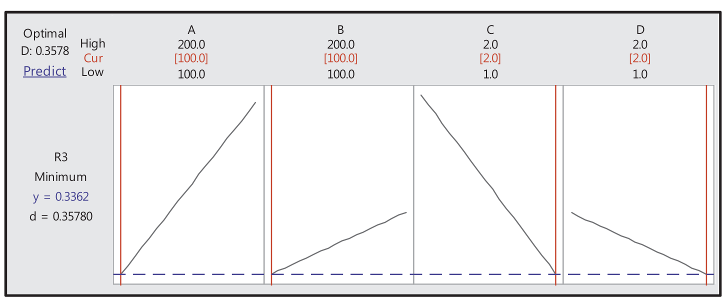
MOPSO and NSGA-III are two examples of evolutionary optimization algorithms, each with several parameters, which should be tuned while they are utilized to solve an optimization problem. The parameters of the MOPSO and NSGA-III algorithms and their levels are



3(a): First response variable as objective function in the MOPSO performance optimization



3(b): Second response variable as objective function in the MOPSO performance optimization



3(c): Third response variable as objective function in the MOPSO performance optimization

Fig. 3. Individual desirability function results to tune the parameters of MOPSO.

Table 8
Results of individual desirability function in tuning the parameters of MOPSO.

Response	Optimal solution				Predicted response	Desirability value
	A	B	C	D		
R1	200	200	1	1	724.054	0.3052
R2	100	–	2	1	43.4131	0.8695
R3	100	100	2	2	0.33623	0.3578

shown in Tables 1 and 2, respectively. In this article, we only consider two levels for each parameter (factor).

5.2. Performance metrics of evolutionary optimization algorithms

In order to evaluate the performance of the MOEAs, i.e., MOPSO and NSGA-III, several performance metrics are used to measure the various features of the algorithms. In this study, we consider the most popular metrics used in the literature which include: the mean ideal distance, spacing, and spread [53,54]. The description of each metric is presented as follows.

The mean ideal distance (MID), proposed by Zitzler and Thiele [60], measures the closeness of each solution in the Pareto frontier to the ideal point which in this study is (0,0,0). This measure is presented in Eq. (18) where n is the number of non-dominated solutions in the Pareto frontier and f_{1i}, f_{2i}, and f_{3i} represent the first, second, and third objective values of the i th non-dominated solution, respectively:

$$MID = \frac{1}{n}\sum_{i=1}^{n}\left(\sqrt{f_{1i}+f_{2i}+f_{3i}}\right) \tag{18}$$

The next performance metric, i.e. spacing, represents the relative distances of consecutive solutions in the Pareto frontier Akhavan Niaki et al. [3]. Eq. (19) shows this metric in which $dist_i = \min_{k\in E\wedge k\neq i}\sum_{m=1}^{M}|f_m^i - f_m^k|$ and $\overline{dist} = \sum_{i=1}^{|n|}\frac{dist_i}{n}$.

$$sp = \sqrt{\frac{1}{|n|}\sum_{i=1}^{|n|}(dist_i - \overline{dist})^2} \tag{19}$$

Note that if all the solutions in the Pareto frontier are equally

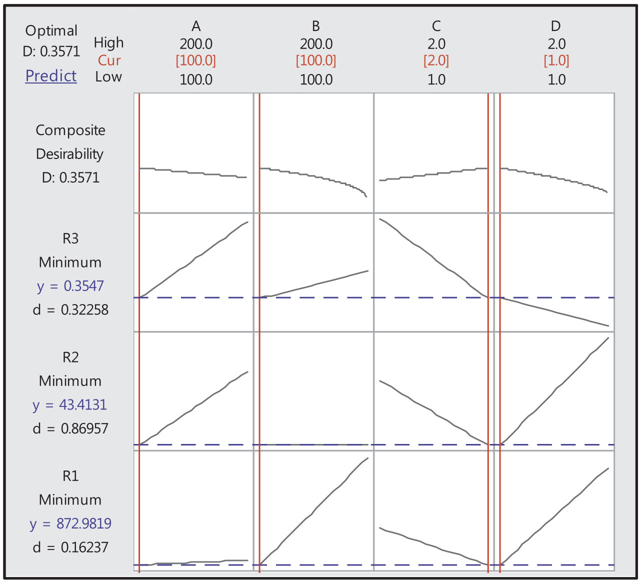


Fig. 4. Composite desirability function approach (multi responses optimization) to tune the parameters of MOPSO.

Table 9
Results of composite desirability function in tuning the parameters of MOPSO.

Response	Factors				Predicted response	Desirability value
	A	B	C	D		
R1	100	100	2	1	872.982	0.68842
R2					43.4131	
R3					0.354677	

spread, then the spacing metric would be equal to zero.

The next metric, spread, is proposed by Deb et al. [12] which measures the spread of the solutions in the Pareto frontier as presented in Eq. (20):

$$\Delta = \frac{\sum_{m=1}^{M} dist_m^e + \sum_{i=1}^{|E|} |dist_i - \overline{dist}|}{\sum_{m=1}^{M} dist_m^e + |E|\overline{dist}} \tag{20}$$

where $dist_i$ is the distance between the neighbour solutions, $\overline{dist}$ is the average distance, $dist_m^e$ is the distance between the extreme solution of the problems and E corresponds to the m th objective function. When the solutions are ideally distributed, the spread metric will be zero.

In this research, the described performance metrics for the evolutionary algorithms are considered as response variables which need to be optimized. Table 3 summaries the properties of each response variable, i.e. the performance metrics used to measure the performance of the MOEAs proposed in this research.

6. Computational results and discussion

In this section, we apply the proposed two-stage approach to tune the parameters (factors) of the algorithms, i.e. MOPSO and NSGA-III, for solving a case study problem. Considering all the performance metrics (response variables) in evaluating the performance of the algorithms, the optimal settings of the parameters (factors) for the algorithms are provided. A full factorial design of experiment is used in the first stage to estimate the regression model, and then, the desirability function approach is applied to find the optimal values of the algorithms' parameters. The two-stage tuning approach is described as follows, in detail.

6.1. Experimental design

To investigate the optimal values of the algorithms' parameters, a full factorial design of experiment is used by considering four parameters (factors), each with two values (levels) for each evolutionary algorithm. We consider 4 factors, each with two levels, resulting in 16 treatment combinations in the experiment for each evolutionary algorithm. The MOPSO and NSGA-III are coded in MATLAB 2014Ra and run for a problem with 50 jobs, which is considered as large size instances. Tables 4 and 5 display the values obtained for each metric for each parameter value, where the algorithms are run 30 times for each parameter. The average of the metrics for both MOPSO and NSGA-III are shown in Tables 4 and 5, respectively.

6.2. Application of the full factorial design of experiment

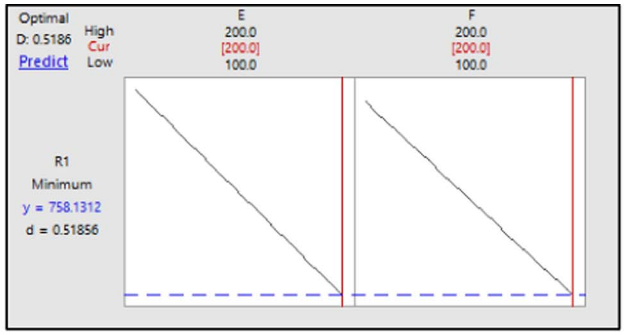
In this section, the full factorial design of experiment is conducted to see which factors are significant, and if they are significant, how they affect the response variables. The other objectives for applying a full factorial design of experiment are to find the estimated regression model for each response variable. In this regard, the effect of factors on each response variable is investigated individually, where the results are summarized as follows.

6.2.1. Performance evaluation of the MOPSO algorithm

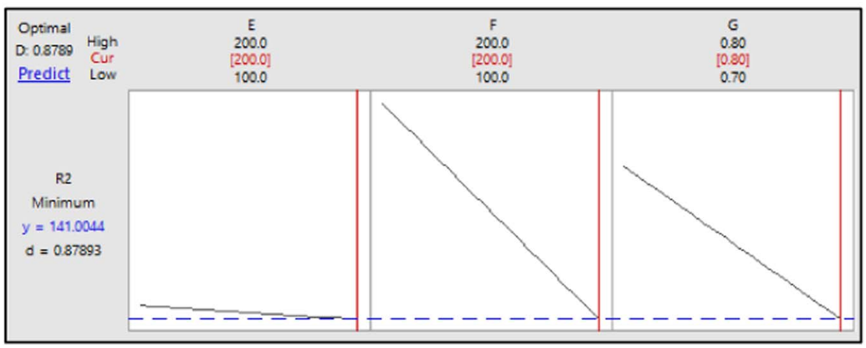
First, the first response variable is investigated. In terms of the factors which affect the performance of MOPSO, among all the factors and their interactions, only factor A, B, C, D, and the AB interaction are obtained to be significant using the full factorial design of experiment. The regression model for R_1 is obtained as: $R_1 = 753 + 0.480A + 1.110B + 34.6C + 70.4D - 0.01023A * B$. The main and interaction plots are presented as Fig. 1(*a* and *b*). According to Fig. 1.1(a), all the main factors are significant; while Fig. 1(b) shows that only the interaction between factor A and B is significant. Fig. 1.2, also shows that only factors A, C, D, and AC are significant. The regression model for R_2,including the significant factor is obtained as: $R_2 = 131.2 - 0.552A - 177.4C + 158.1D + 0.820A * C$. The main effect and interaction plots for the second response variable are displayed in Fig. 1.2. To simplify the problem, only the significant factors and interactions are considered to create the plots drawn in Fig. 1.2. According to Fig. 1.3, considering the third response variable in the full factorial design of experiment shows that only the main factors are significant. In this case, the regression model for R_3 includes factors A, B, C, and D. The regression model for R_3 is presented as: $R_3 = 0.410 + 0.000516A + 0.000187B - 0.0538C - 0.0184D$. Since there is no significant interaction, only the main factors are significant, which are presented in Fig. 1.3. In all the investigations, the assumption of the full factorial design of experiment are tested and the results show no violation of the assumptions. The summary of results obtained from the full factorial design of experiment is presented in Table 6. It should be noted that there is no optimal value for the insignificant factors, for example, factor B in the response variable 2; therefore, the "dash" symbol is used in some tables.

6.2.2. Performance evaluation of the NSGA-III algorithm

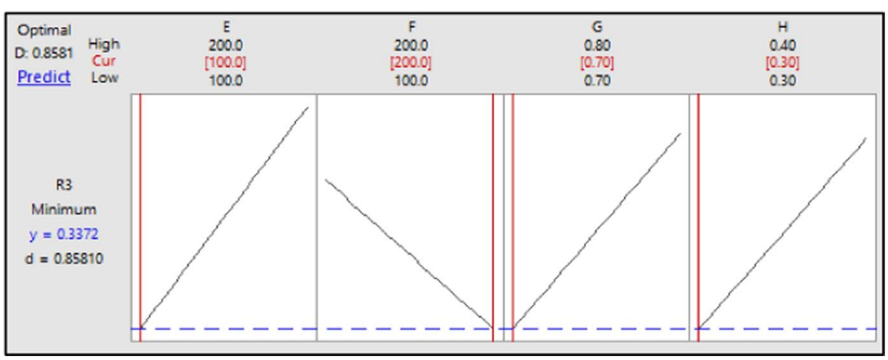
Investigating the effect of the NSGA-III parameters on the first response variable shows that only factors E and F as well as their interactions are significant (Fig. 2.1). The regression model provided by the full factorial design of experiment is formulated as follows: $R_1 = 826 + 0.87E + 0.95F - 0.01078E * F$. Considering the second response variable, the significant factors are E, F, G, and their interactions, except E∗F∗H (Fig. 2.2). Therefore, the regression model for R_2 is formulated as $R_2 = -1361 - 2.8E + 6.2F + 2416G + 0.0498E * F + 3.4E * G - 10.3F * G$. Our investigation also shows that all factors and their interactions are significant considering the third response variable (Fig. 2.3). The regression model for R_3 is obtained as:



5(a): First response variable as an objective function in the NSGA-III performance optimization



5(b): Second response variable as an objective function in NSGA-III performance optimization



5(c): Third response variable as an objective function in NSGA-III performance optimization

Fig. 5. Individual desirability function results to tune the parameters of NSGA-III.

Table 10
Results of individual desirability function in tuning the parameters of NSGA-III.

Response	Optimal solution				Predicted response	Desirability value
	E	F	G	H		
R1	200	200	–	–	758.1312	0.51856
R2	200	200	0.8	–	141.0044	0.87893
R3	100	200	0.7	0.3	0.3372	0.85810

$R_3 = 0.24 - 0.0034E - 0.00361F + 0.63G - 4.4H + 0.000019E*F - 0.0005E*G + 0.035E*H + 0.0053F*H + 5.9G*H - 0.000029E*F*H - 0.035E*G*H$. Table 7 presents a summary of results obtained from the full factorial design of experiment application when each response variable is investigated individually. The significant factors and their recommended levels to obtain the optimal response variable, i.e., performance metric of the NSGA-III algorithm are presented in Table 7.

6.3. Desirability function approach to tune the parameter of evolutionary optimization algorithms

Considering the regression models obtained from full factorial design of experiment as the objective functions, the individual and composite desirability function approaches are used to find the optimal setting of each factor. First, the individual desirability function approach is used and each response variable is optimized individually, and then, the composite desirability function approach is used, while all objective functions (response variables) are optimized simultaneously.

The results of the individual desirability function approach are presented in Fig. 3 for the MOPSO algorithm, in which the optimal values of the factors are presented. The summary of results obtained from the individual desirability function approach is presented in

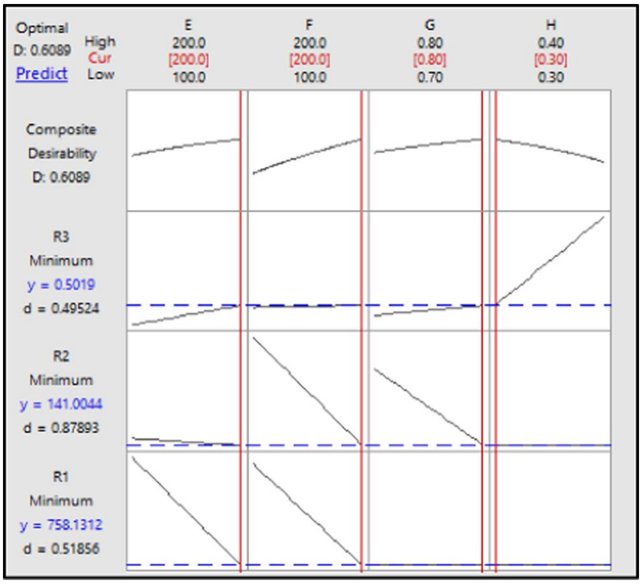


Fig. 6. Composite desirability function approach (multi response optimization) to tune the parameters of NSGA-III.

Table 11
Results of composite desirability function in tuning the parameters of NSGA-III.

Response	Factors				Predicted response	Desirability value
	E	F	G	H		
R1	200	200	0.8	0.3	0.5019	0.6089
R2					141.0044	
R3					758.1312	

Table 8.

After investigating each response variable as an objective function individually, all response variables are optimized using the desirability function approach, while three response variables are considered as objective functions simultaneously. The results are presented in Fig. 4, where the optimal values of the factors are presented. In Fig. 4, all the response variables are optimized simultaneously. Comparing the results obtained from the full factorial design of experiment, the individual desirability function approach, and the composite desirability function approach show that although optimizing each response variable individually will provide a better result for each response variable, the optimal parameter values will be different when each response variable is optimized individually. For example, considering the first response variable as an objective function, the optimal values of factors A, B, C, and D are obtained as 200, 200, 1, and 1, respectively. Also, when response variable 3 is considered as a response variable, the optimal values are different. Considering all the response variable as objective functions simultaneously in the composite desirability function method, generates one general value for all the parameters of the algorithms, which leads to an optimal value of all the response variables. The summary of the results obtained from the composite desirability function approach is presented in Table 9.

Additionally, each response variable is considered individually to find the optimal values of the parameters of NSGA-III algorithm. The results are provided in Fig. 5, and summarized in Table 10.

All the response variables are also considered in the composite desirability function optimization approach to find the optimal values of the parameters, while all the response variables are optimized simultaneously. The results are presented in Fig. 6 and Table 11.

Note that the proposed hybrid parameter tuning approach can also be applied in the parameter tuning of the evolutionary optimization algorithm in which more than four parameters, i.e., factors, are defined as significant parameters of the evolutionary optimization algorithm. It can also be utilized when there are more than three performance metrics, i.e., response variables, to be optimized.

7. Conclusion and future research directions

The paper proposed a methodology to tune the parameters of the evolutionary optimization algorithms based on the desirability function approach. Two popular multi-objective evolutionary algorithms, MOPSO and NSGA-III were investigated using the proposed approach. The optimal values of the parameters of the evolutionary optimization algorithms were provided, while all performance metrics are optimized simultaneously. First, a full factorial design of experiment was applied to find the approximate regression model for each response variable, i.e., the performance metrics of evolutionary optimization algorithms. The regression model included the significant parameters of the algorithms which affected the performance of the algorithms in solving an optimization problem. Then, using the regression model as an objective function, the composite desirability function approach was utilized to find the optimal values of the parameters of the algorithms, where all the response variables, were optimized. A case study problem of multiple objectives single machine scheduling problem was developed to evaluate the efficiency and effectiveness of the proposed approach in a real case problem. MOPSO and NSGA-III were used to solve the developed case study problem. The full factorial design of experiment was considered to run the algorithms while changing the values of the algorithms parameters in each run. Finally, several parameters of the algorithms were tuned using the composite desirability function approach.

Results of the proposed hybrid composite desirability function approach were compared with the individual desirability function approach and a full factorial design of experiment, in which each performance metric is optimized individually while tuning the parameter of algorithm. The most significant benefit of the proposed composite desirability function approach is that it simultaneously optimized all performance metrics of the evolutionary optimization algorithm while tuning its parameters. In other words, instead of having different optimal settings of parameters of the evolutionary algorithm, while each performance metric is optimized individually, a general optimal setting of parameters can be obtained which optimizes all performance metrics of the evolutionary optimization algorithm. The proposed hybrid parameter tuning approach can be utilized to tune the parameters of optimization evolutionary algorithms, regardless of the number of objectives in the optimization problem.

As a recommendation for the future research, the proposed method can be applied for different engineering and business multi-objective and single-objective optimization problems. Furthermore, other evolutionary optimization algorithms can be investigated to show other applications of the proposed parameter tuning approach. Besides, the proposed parameter tuning approach can be utilized in the case problems in which more than three performance metrics need to be optimized in the performance evaluation of the evolutionary optimization algorithm, or/and there are more than four parameters of the evolutionary optimization algorithm to be tuned. In addition, a sensitivity analysis of considering different weights for each performance metric of the evolutionary optimization algorithms can be considered as an extension of the proposed approach.

Acknowledgement

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

References

[1] A.M. Aboutaleb, L. Bian, A. Elwany, N. Shamsaei, Scott M. Thompson, G. Tapia, Accelerated process optimization for laser-based additive manufacturing by leveraging similar prior studies, IISE Trans. (2016), http://dx.doi.org/10.1080/0740817X.2016.1189629.

[2] B. Akay, D. Karaboga, Parameter tuning for the artificial bee colony algorithm, International Conference on Computational Collective Intelligence, Springer, 2009, pp. 608–619.

[3] S.T. Akhavan Niaki, M. Malaki, M.J. Ershadi, A comparative study of four evolutionary algorithms for economic and economic-statistical designs of MEWMA control charts, J. Optim. Ind. Eng. 9 (2011) 1–13.

[4] N. Alikar, S.M. Mousavi, R.A.R. Ghazilla, M. Tavana, E.U. Olugu, A bi-objective multi-period series-parallel inventory-redundancy allocation problem with time value of money and inflation considerations, Comput. Ind. Eng. 104 (2017) 51–67.

[5] N. Alikar, S.M. Mousavi, R.A. Raja Ghazilla, M. Tavana, E.U. Olugu, Application of the NSGA-II algorithm to a multi-period inventory-redundancy allocation problem in a series-parallel system, Reliab. Eng. Syst. Saf. 160 (2017) 1–10.

[6] A. Azhari, C. Schindler, B. Li, Effect of waterjet peening on aluminum alloy 5005, Int. J. Adv. Manuf. Technol. 67 (2013) 785–795.

[7] M. Azzeh, A.B. Nassif, S. Banitaan, F. Almasalha, Pareto efficient multi-objective optimization for local tuning of analogy-based estimation, Neural Comput. Appl. 27 (8) (2016) 2241–2265.

[8] B. Balamugundan, L. Karthikeyan, V.S. Senthilkumar, Multi characteristics optimization during milling of friction stir processed glass fiber reinforced plastic composites, Procedia Eng. 38 (2012) 1276–1285.

[9] J.F. Boulet, A. Gharbi, J.P. Kenné, Multi-objective optimization in an unreliable failure-prone manufacturing system, J. Qual. Maint. Eng. 15 (4) (2009) 397–411.

[10] C.A.C. Coello, G.T. Pulido, M.S. Lechuga, Handling multiple objectives with particle swarm optimization, IEEE Trans. Evolut. Comput. 8 (3) (2004) 256–279.

[11] B. Crawford, R. Soto, E. Monfroy, W. Palma, C. Castro, F. Paredes, Parameter tuning of a choice-function based hyperheuristic using particle swarm optimization, Expert Syst. Appl. 40 (5) (2013) 1690–1695.

[12] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evolut. Comput. 6 (2) (2002) 182–197.

[13] G. Derringer, R. Suich, Simultaneous optimization of several response variables, J. Qual. Technol. 12 (4) (1980) 214–219.

[14] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, MHS'95, IEEE, 1995, pp. 39–43.

[15] A.E. Eiben, S.K. Smit, Parameter tuning for configuring and analyzing evolutionary algorithms, Swarm Evolut. Comput. 1 (1) (2011) 19–31.

[16] S. Gohari, N. Salmasi, Flexible flowline scheduling problem with constraints for the beginning and terminating time of processing of jobs at stages, Int. J. Comput. Integr. Manuf. 28 (10) (2015) 1092–1105.

[17] D.E. Golberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addion Wesley, 1989, p. 102.

[18] R.L. Graham, E.L. Lawler, J.K. Lenstra, A.R. Kan, Optimization and approximation in deterministic sequencing and scheduling: a survey, Ann. Discrete Math. 5 (1979) 287–326.

[19] E. Hajizadeh, M. Mahootchi, A. Esfahanipour, M.M. Khah, A new NN-PSO hybrid model for forecasting Euro/Dollar exchange rate volatility, Neural Comput. Appl. (2015), http://dx.doi.org/10.1007/s00521-015-2032-7.

[20] E. Harrington, The desirability function, Ind. Qual. Control 21 (10) (1965) 494–498.

[21] Y. Hashemi, H. Shayeghi, M. Moradzadeh, Design of dual-dimensional controller based on multi-objective gravitational search optimization algorithm for amelioration of impact of oscillation in power generated by large-scale wind farms, Appl. Soft Comput. (2016), http://dx.doi.org/10.1016/j.asoc.2016.12.036.

[22] K. Hassani, K. Jafarian, An intelligent method for breast cancer diagnosis based on fuzzy ART and metaheuristic optimization, in: XIV Mediterranean Conference on Medical and Biological Engineering and Computing (MEDICON 2016), Paphos, Cyprus, 2016.

[23] M. Heydari, K.M. Sullivan, E.A. Pohl, Optimal allocation of testing resources in reliability growth, in: Proceedings of the 2014 Industrial and Systems Engineering Research Conference, 2014, pp. 3423–3428.

[24] A. Ibrahim, S. Rahnamayan, M.V. Martin, K. Deb, Elite NSGA-III: An improved evolutionary many-objective optimization algorithm, in: IEEE Congress on Evolutionary Computation (CEC), IEEE, 2016, pp. 973–982.

[25] N. Iwasaki, K. Yasuda, G. Ueno, Dynamic parameter tuning of particle swarm optimization, IEEJ Trans. Electr. Electron. Eng. 1 (4) (2006) 353–363.

[26] R. Jafari-Marandi, H. Mengqi, A.O. OluFemi, A distributed decision framework for building clusters with different heterogeneity settings, Appl. Energy 165 (2016) 393–404.

[27] H. Jain, K. Deb, An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: handling constraints and extending to an adaptive approach, IEEE Trans. Evolut. Comput. 18 (4) (2014) 602–622.

[28] V. Kayvanfar, M. Zandieh, The economic lot scheduling problem with deteriorating items and shortage: an imperialist competitive algorithm, Int. J. Adv. Manuf. Technol. 62 (5) (2012) 759–773.

[29] V. Kayvanfar, M. Zandieh, E. Teymourian, An intelligent water drop algorithm to identical parallel machine scheduling with controllable processing times: a just-in-time approach, Comput. Appl. Math. 36 (1) (2017) 159–184.

[30] S. Kirkpatrick, Optimization by simulated annealing: Quantitative studies, J. Stat. Phys. 34 (5–6) (1984) 975–986.

[31] N.N. Korra, M. Vasudevan, K.R. Balasubramanian, Multi-objective optimization of activated tungsten inert gas welding of duplex stainless steel using response surface methodology, Int. J. Adv. Manuf. Technol. 77 (1–4) (2014) 67–81.

[32] E.L. Lawler, A "pseudopolynomial" algorithm for sequencing jobs to minimize total tardiness, Ann. Discrete Math. 1 (1977) 331–342.

[33] C. Li, Q. Xiao, Y. Tang, L. Li, A method integrating Taguchi, RSM and MOPSO to CNC machining parameters optimization for energy saving, J. Clean. Prod. 135 (2016) 263–275.

[34] Z. Michalewicz, S.J. Hartley, Genetic algorithms + data structures = evolution programs, Math. Intell. 18 (3) (1996) 71.

[35] S. Mirjalili, Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, Neural Comput. Appl. 27 (4) (2016) 1053–1073.

[36] M. Mobin, M. Mobin, Z. Li, Multi-response optimization of cavitation peening parameters for improving fatigue performance using the desirability function approach, Int. J. Appl. Decis. Sci. 9 (2) (2016) 156–181.

[37] S.M. Mousavi, N. Alikar, S.T.A. Niaki, A. Bahreininejad, Optimizing a location allocation-inventory problem in a two-echelon supply chain network: a modified fruit fly optimization algorithm, Comput. Ind. Eng. 87 (2015) 543–560.

[38] S.M. Mousavi, J. Sadeghi, S.T.A. Niaki, M. Tavana, A bi-objective inventory optimization model under inflation and discount using tuned Pareto-based algorithms: NSGA-II, NRGA, and MOPSO, Appl. Soft Comput. 43 (2016) 57–72.

[39] T.E. Murphy, K.L. Tsui, J.K. Allen, A review of robust design methods for multiple responses, Res. Eng. Design 15 (4) (2005) 201–215.

[40] B. Naderi, S. Gohari, M. Yazdani, Hybrid flexible flowshop problems: models and solution methods, Appl. Math. Model. 38 (24) (2014) 5767–5780.

[41] S. Naka, T. Genji, T. Yura, Y. Fukuyama, Practical distribution state estimation using hybrid particle swarm optimization, in: Power Engineering Society Winter Meeting, vol. 2, IEEE, 2001, pp. 815–820.

[42] S.H.R. Pasandideh, S.T.A. Niaki, S.M. Mousavi, Two metaheuristics to solve a multi-item multiperiod inventory control problem under storage constraint and discounts, Int. J. Adv. Manuf. Technol. (2013) 1–14.

[43] R. Ramanujam, R. Raju, N. Muthukrishnan, Taguchi multi-machining characteristics optimization in turning of Al-15% SiCp composites using desirability function analysis, J. Stud. Manuf. 1 (2/3) (2010) 120–125.

[44] E. Rashidi, M. Parsafard, H. Medal, X. Li, Optimal traffic calming: A mixed-integer bi-level programming model for locating sidewalks and crosswalks in a multimodal transportation network to maximize pedestrians' safety and network usability, Transp. Res. Part E: Logist. Transp. Rev. 91 (2016) 33–50.

[45] M. Reihaneh, D. Karapetyan, An efficient hybrid ant colony system for the generalized traveling salesman problem, Algorit. Operat. Res. 7 (2012) 22–29.

[46] M. Salari, M. Reihaneh, M.S. Sabbagh, Combining ant colony optimization algorithm and dynamic programming technique for solving the covering salesman problem, Comput. Ind. Eng. 83 (2015) 244–251.

[47] Y. Shi, R.C. Eberhart, Empirical study of particle swarm optimization, in: Proceedings of the 1999 Congress on Evolutionary Computation, CEC 99, vol. 3, IEEE, 1999.

[48] S.K. Smit, A. Eiben, Parameter tuning of evolutionary algorithms: Generalist vs. specialist, in: European Conference on the Applications of Evolutionary Computation, Springer, 2010, pp. 542–551.

[49] S.K. Smit, A.E. Eiben, Comparing parameter tuning methods for evolutionary algorithms, in: 2009 IEEE Congress on Evolutionary Computation, IEEE, 2009, pp. 399–406.

[50] N. Srinivas, K. Deb, Muiltiobjective optimization using nondominated sorting in genetic algorithms, Evolut. Comput. 2 (3) (1994) 221–248.

[51] Y.-X. Su, R. Chi, Multi-objective particle swarm-differential evolution algorithm, Neural Comput. Appl. (2015), http://dx.doi.org/10.1007/s00521-015-2073.

[52] G. Taguchi, G. Taguchi, System of experimental design; engineering methods to optimize quality and minimize costs, 1987 (No. 04; QA279, T3).

[53] M. Tavana, M.R. Kazemi, A. Vafadarnikjoo, M. Mobin, An artificial immune algorithm for ergonomic product classification using anthropometric measurements, Measurement 94 (2016) 621–629.

[54] M. Tavana, Z. Li, M. Mobin, M. Komaki, E. Teymourian, Multi-objective control chart design optimization using NSGA-III and MOPSO enhanced with DEA and TOPSIS, Expert Syst. Appl. 50 (2016) 17–39.

[55] E. Teymourian, V. Kayvanfar, G.M. Komaki, M. Khodarahmi, An enhanced intelligent water drops algorithm for scheduling of an agile manufacturing system, Int. J. Inf. Technol. Decis. Making 15 (2) (2016) 239–266.

[56] A. Vafadarnikjoo, S.M.A.K. Firouzabadi, M. Mobin, A. Roshani, A meta-heuristic approach to locate optimal switch locations in cellular mobile networks, in: Proceedings of the International Annual Conference of the American Society for Engineering Management (ASEM2015), Indiana, USA, 2015.

[57] B. Vahdani, D. Veysmoradi, N. Shekari, S.M. Mousavi, Multi-objective, multi-period location-routing model to distribute relief after earthquake by considering emergency roadway repair, Neural Comput. Appl. (2016), http://dx.doi.org/10.1007/s00521-016-2696-7.

[58] M.G. Varzaneh, R. Sawhney, H. Shams, A. Asadinejad, Distribution of load change in industrial demand: a DOE approach, IN: Innovative Smart Grid Technologies Conference (ISGT), 2016 IEEE Power & Energy Society, September, IEEE, 2016, pp. 1–5.

[59] A.J. Yu, J. Seif, Minimizing tardiness and maintenance costs in flow shop scheduling by a lower-bound-based GA, Comput. Ind. Eng. 97 (2016) 26–40.

[60] E. Zitzler, L. Thiele, Multiobjective optimization using evolutionary algorithms-a comparative case study, Parallel Problem Solving from Nature-PPSN V, Springer, 1998, pp. 292–301.