Composite Desirability Function (CDF) Approach for Evolutionary Algorithm Parameter Tuning

Mohammad Komaki  
Electrical Engineering and Computer Science Department  
Case Western Reserve University  
Cleveland, OH 44120, USA

Behnam Malakooti  
Electrical Engineering and Computer Science Department  
Case Western Reserve University  
Cleveland, OH 44120, USA

Abstract

Parameters values of evolutionary algorithms have significant effect on the performance of algorithms. The process of parameter setting is important and this subject has been investigated in the optimization literature using different design of experiment methods, such as the Taguchi method, the full factorial design of experiment, the surface response methodology, etc. All mentioned approaches considered only the individual response variable, i.e., a performance metric for evaluating algorithm, in finding the optimal setting for the parameters. However, there are multiple performance metrics that should be taken into consideration when tuning a parameter. In this research, a new approach to tune evolutionary algorithms parameters is proposed which simultaneously optimizes all performance metrics of the algorithm and provides the optimal setting for the parameters of the algorithm. To show the application of the proposed approach, a case study is developed based on a multi-objective optimization problem. The developed problem is solved using multi-objective particle swarm optimization algorithm (MOPSO) which considers different setting of algorithm parameters, i.e., scenario, using full factorial design of experiment. In each scenario, different performance metrics are calculated as response variables. Then, the desirability function approaches are applied to provide the optimal setting of the parameters when all response variables are optimized.

Keywords
Algorithm parameter tuning, Desirability function approach, MOPSO, Multi-objective single machine

1. Introduction

One of the challenging, yet not appropriately investigated, questions in developing an evolutionary algorithm is to find the optimal setting of the parameters of the algorithm. This process, which is known as parameter tuning in the evolutionary algorithm literature, has always been a challenge in developing a new algorithm as well as application of developed algorithm in different optimization problems. Evolutionary algorithms or metaheuristic algorithms which mimic the natural processes in the world are general framework that can be applied to all optimization problems while other algorithms such as heuristic algorithms are problem specific. Each evolutionary algorithm starts with a group of initial solution (or an individual solution) and iterates several operations on the solutions until stopping condition is met. Some of the well-known metaheuristic algorithms are Genetic Algorithm (Goldberg, 1989), Simulated Annealing (Kirkpatrick et al., 1983), Particle Swarm Optimization (Eberhart and Kennedy, 1995), etc. Each of these algorithms are inspired from behavior different
phenomena and each of them have its own parameters which needs to be set. For instance, Genetic algorithm and Particle Swarm Optimization are population based algorithms, therefore, one needs to set the population size to solve the problem at hand. Note that the performance of one algorithm highly depends on the parameters which one setting for a problem may not be a proper setting for another problem. Therefore, the parameter tuning should be problem specific.

In the literature, there are few approaches to tune the parameters of different evolutionary algorithms for a single objective problems. Akay and Karaboga (2009) investigate the performance of the artificial bee colony algorithm by analyzing the effect of control parameters. A comparison of parameter tuning methods for evolutionary algorithm is presented by Smit and Eiben (2009) and Eiben et al. (2011). Crawford et al. (2013) applied a particle swarm optimization to tune the parameter of a choice function based hyper-heuristic. Smit et al. (2010) applied the relevance estimation and value calibration method, called REVAC, to find the good evolutionary algorithm parameter values. Iwasaki et al. (2006) proposed a dynamic parameter tuning for the particle swarm optimization algorithm. The feasibility and advantages of their proposed adaptive PSO algorithm are verified through numerical simulations using some typical global optimization problems. Most of studies heavily rely on the parameter tuning of other studies. For instance, Rashidi et al. (2016) proposed Simulated Annealing algorithm to solve locating sidewalks and crosswalks in transportation network and used the parameter values suggested by other researchers for their algorithm. Therefore, tuning parameters is very important.

In addition to the parameter tuning of the evolutionary algorithm, different approaches have been also applied in the literature to tune the parameter of the machine learning methods. Some can be mentioned as Imbault and Lebart (2004), Arcuri and Fraser (2011), Gold et al. (2005), He and Ounis (2003), Kulkarni et al. (2004), and Wang et al. 2006. There are also some studies such as Taguchi method (Taguchi, 1987) which applied different design of experiment methods to tune the parameter of the evolutionary algorithm. Taguchi method has been criticized to be inefficient.

From the reviewed literature, it can be revealed that almost all parameter tuning approaches only consider one performance metric of the algorithm and find the optimal setting of the parameters. However, there are multiple performance metrics for an evolutionary algorithm to solve a multi-objective optimization problem, such as mean ideal distance, spacing, and spread that need to be taken into consideration when the parameters of an algorithm are tuning. This necessitates application of multi-response optimization approaches in parameter tuning. In this research, we proposed the application of composite desirability function approach, which optimize all performance metrics of the algorithm, while providing the optimal setting of the parameters of the algorithm. In this regard, obtaining a regression model for each performance metric is required. In the proposed approach, we applied full factorial design of experiment to obtain the significant parameters of the algorithm, which play a role in the regression model.

The rest of this paper is organized as follows. Section 2 is devoted to the developed methodology to find the optimal parameters of evolutionary algorithms. Section 3 presents a case study and solving it using Multi-objective Particle Swarm Optimization. Also, this section covers the considered performance metrics and parameters of the developed evolutionary algorithm. The detail of application of the proposed methodology is covered in this section too. Finally, Section 4 presents conclusion and future work.

2. Proposed methodology

To obtain the optimal setting of the parameters for the evolutionary algorithm, there are some performance metrics for the performance evaluation of the evolutionary algorithm. Each parameter can be considered as a factor, each with different predefined levels, and the performance metrics can be considered as response variables. Considering all combinations of factor levels, a full factorial design of experiment can be conducted to investigate the effect of each factor on response variable, the significant factors, and the approximate regression model for each response variable. Details of the full factorial design description and applications can be found in Mobin et al., 2015. Then considering each regression model as an objective function, the desirability function approach can be applied to optimize all response variables simultaneously, and obtain the optimal setting for all factors. A brief description of desirability function approach is presented as follows.

Desirability function approach (DFA) is first introduced by Harrington in 1965 and the extended later by Derringer and Suich (1980). This method is a search-based optimization method which optimizes multiple response variables, individually and simultaneously, to find the optimum input variable settings (Mobin et al., 2016).

Analysis of multiple response variables includes first creating a mathematical model, known as regression model, for each response variable. Then, a set of optimized factors can be obtained which optimizes all responses. Solving multi-response optimization problems starts with applying a technique for integrating multiple responses into a dimensionless function, called the overall desirability function (D). The approach is to first convert each response
(y_i) into a dimensionless function, known as 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). In the desirability function approach, there is a positive number, known as the weight factor (w), on which the shape of the desirability function for each response depends (Mobin et al., 2016). For the sake of simplicity, the weights for response variables are considered equal to one.

The individual desirability functions can be calculated based on the optimization functions, i.e. maximization or minimization using Equations (1)-(3). If the target (T_i) for the response y_i is a maximum value, the desirability is based on Equation (1). If the target is to get the minimum value, the desirability is based on Equation (2). Finally, if the target is located between the lower (L_i) and upper (U_i) limits, the desirability is based on Equation (3).

\[ d_i = \begin{cases} 
0 & y_i < L_i \\
\left(\frac{y_i - L_i}{U_i - L_i}\right)^w & L_i \leq y_i \leq T_i \\
1 & y_i > T_i 
\end{cases} \]  
\quad (1) 

\[ d_i = \begin{cases} 
1 & y_i < L_i \\
\left(\frac{T_i - y_i}{T_i - L_i}\right)^w & L_i \leq y_i \leq T_i \\
0 & y_i > T_i 
\end{cases} \]  
\quad (2) 

\[ d_i = \begin{cases} 
\left(\frac{y_i - L_i}{U_i - L_i}\right)^w & T_i \leq y_i \leq U_i \\
0 & y_i > U_i 
\end{cases} \]  
\quad (3)

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

\[ D = (d_1^w_1 \times d_2^w_2 \times d_3^w_3 \times \ldots \times d_n^w_n)^{\frac{1}{\sum w_i}} = \prod_{i=1}^{n} d_i^{w_i} \]  
\quad (4)

More details about the desirability function approach and its applications are presented in (Mobin et al., 2016) in which they applied desirability function approach to optimize the multi-response cavitation process.

### 3. Application of the proposed approach in a case study

#### 3.1. Case study description

In this section, we study tri-objective single machine to minimize 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 the machine one 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. Let \pi be sequence of jobs where \pi(1) represents the job in the first position, then the objectives of sequence \pi can be computed as following.

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

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

Eqs. (5) and (6) compute the completion time and tardiness of job in position i of the sequence \pi, 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, makespan of the sequence \pi is:

\[ C_{\text{max}} = C(\pi(N)) \]  
\quad (7)

The total weighted completion time (TWC) of the sequence \pi is as following.

\[ TWC = \sum_{i=1}^{N} w_{\pi(i)}\pi(i)) \]  
\quad (8)

Total weighted tardiness (TWT) of the sequence \pi is as following.
Using three-field problem classification $d|p|y$ of Graham et al. (1979), the addressed problem can be presented as $1|d,S|C_{\text{max}}, \text{TWT}, \text{TWC}$ and it has been that each of the considered objective, for example $1|d,S|\text{TWT}$, is strongly NP-hard, see Lawler (1977).

### 3.2 Evolutionary algorithm to solve the case problem

There are several MOEAs to solve the described problem in the pervious section. MOPSO, NSGA-II, and NSGA-III are the most commonly used algorithm that have been used to solve the MOEAs algorithm (Mobin et al., 2015; Li et al., 2016; Tavana et al., 2016). For example, Mobin et al. 2015 applied NSGA-II to optimize the multi-objective X-bar control chart design. Li et al. (2016) applied NSGA-II to optimize the multi-objective reliability growth plan problem. Tavana et al. (2016) applied NSGA-III and MOPSO to optimize the multi-objective design of economical statistical control chart problem. In this research, we apply MOPSO (Coello et al., 2004) to find the Pareto optimal frontier. MOPSO is a population-based metaheuristic algorithm inspired from social behaviour of birds. Each bird (or particle) represents a solution of the problem. The algorithm starts with initial population of particles which is randomly generated. The number of particles (solutions) $N_{\text{pop}}$ is one parameter that needs to be set. MOPSO is an iterative algorithm, that is, it iterates $I_{\text{max}}$ times by performing same operations at each iteration. The location of each particle $b$ at each iteration $t$ of the algorithm is presented by $s^t_b$ and its velocity is represented by $v^t_b$. In this paper, the initial position and velocity of each particle $b = 1, 2, ..., N_{\text{pop}}$ is generated randomly and then the largest rule is applied on the $s^t_b$ 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 REP (Coello et al., 2004). 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 well-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 presented by $p_{\text{Best}}^t_b$ and the best position found by the swarm is presented by $g_{\text{Best}}^t$. Usually, $g_{\text{Best}}^t$ 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 $p_{\text{Best}}^t_b$ and $g_{\text{Best}}^t$. At each iteration $t$ of the algorithm, the velocity of the particle $b$ is updated as following.

$$v^t_b = w^t_b \cdot v^{t-1}_b + C_1 \cdot r_1 \cdot (p_{\text{Best}}^t_b - s^{t-1}_b) + C_2 \cdot r_2 \cdot (g_{\text{Best}}^t - s^{t-1}_b)$$

(10)

where $r_1$ and $r_2$ random numbers between 0 and 1, and $C_1$ and $C_2$ are constants to control the effects of $p_{\text{Best}}^t_b$ and $g_{\text{Best}}^t$, respectively. Note that $w^t_b$ in Eq. (11), called the inertia weight of the particle $b$, controls the effect of the particles velocity at the previous iteration. Shi and Eberhart (1999) and Naka et al. (2001) suggested using linearly decreasing weight as presented below.

$$w^t_b = \frac{w_{\text{max}} - w_{\text{min}}}{I_{\text{max}}} \cdot t$$

(11)

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

$$s^t_b = s^{t-1}_b + v^t_b$$

(12)

### 3.3. Performance measurements of the algorithm

In order to evaluate the performance of MOPSO, several metrics can be used to measure various features of the algorithm (Tavana et al., 2016). In this study, we consider widely used metrics which are mean ideal distance, spacing, and spread.

The mean ideal distance (MID), proposed by Zitzler (1999), 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. (9) 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.

$$\text{MID} = \frac{1}{n} \sum_{i=1}^{n} (\sqrt{f_{1i}^2 + f_{2i}^2 + f_{3i}^2})$$

(13)
The next performance metric, spacing, represents the relative distances of consecutive solutions in the Pareto frontier (Akhavan Niaki et al., 2011). Eq. (14) represents this metric where \( \text{dist}_l = \min_{k \in E \cup \Lambda k \in e} \sum_{m=1}^{M} |f_{m} - f_{l}^{k}| \) and 
\[
\text{dist} = \frac{1}{n} \sum_{i=1}^{n} \text{dist}_i.
\]

Note that If all solutions in the Pareto frontier are equally spread, then the spacing metric would be equal to zero.
The next metric, spread, is proposed by Deb et al. (2002) and measures the spread of the solutions in the Pareto frontier as presented in Eq. (15).
\[
\Delta = \frac{\sum_{m=1}^{M} \text{dist}_m^e + \sum_{m=1}^{M} |\text{dist}_m - \text{dist}|}{\sum_{m=1}^{M} \text{dist}_m^e + |E| \text{dist}}
\]
where \( \text{dist}_i \) is the distance between neighbour solutions, \( \text{dist} \) is the average distance, \( \text{dist}_m^e \) is the distance between the extreme solution of the problems and and \( E \) corresponding 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 algorithm are considered as response variables, which need to be optimized. Table 1 summarizes the properties of each response variable.

<table>
<thead>
<tr>
<th>Table 1: The response variables (performance evaluation) for the MOPSO parameter tuning experiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1: MID</td>
</tr>
<tr>
<td>Goal</td>
</tr>
<tr>
<td>Target</td>
</tr>
</tbody>
</table>

### 3.4. Parameters of the evolutionary algorithm

As mentioned in Section 3.2, MOPSO has several parameters that need to be set at their optimal level, before the MOPSO utilized to solve the multi-objective optimization problem. In this research, we only investigate four parameters as presented in Table 2. For the sake of simplicity, we only consider two levels for each factor.

<table>
<thead>
<tr>
<th>Table 2: The factors (MOPSO parameter) for the MOPSO parameter tuning experiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factors</td>
</tr>
<tr>
<td>A</td>
</tr>
<tr>
<td>B</td>
</tr>
<tr>
<td>C</td>
</tr>
<tr>
<td>D</td>
</tr>
</tbody>
</table>

### 3.3. Experimental design and results

In order to investigate the best parameter setting of MOPSO algorithm, we conducted a full factorial design of experiment presented by Mobin et al. (2015) and Aboutaleb et al. (2016). Considering 4 factors each with two levels, there are 16 treatment combinations in this experiment. The MOPSO is coded in Matlab 2014Ra and run for a problem with 50 jobs which is considerably large size instances and it is convincing that if the performance of the algorithm is good in this configuration, it will perform well in other problems with different sizes. It should be mentioned that in the future investigation of the proposed tuning approach, the developed case problems can be solved using other evolutionary algorithms such as NSGA-II (Li et al., 2016), NSGA-III (Tavana et al. 2016), ant colony algorithm (Vafadarnikjoo et al., 2015), imperialist competitive algorithm (Borghei et al., 2015), and general variable neighborhood search algorithm (Komaki et al, 2015).

The following table presents the metrics for each parameter setting. Note that for each parameter setting, MOPSO is run 30 times and average of each metric is presented in the Table 3.
Table 3: Factors combinations and response variables

<table>
<thead>
<tr>
<th>Factors (algorithm parameters)</th>
<th>Response variables</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R1</td>
</tr>
<tr>
<td>Run 1</td>
<td>100</td>
</tr>
<tr>
<td>Run 2</td>
<td>200</td>
</tr>
<tr>
<td>Run 3</td>
<td>100</td>
</tr>
<tr>
<td>Run 4</td>
<td>100</td>
</tr>
<tr>
<td>Run 5</td>
<td>100</td>
</tr>
<tr>
<td>Run 6</td>
<td>200</td>
</tr>
<tr>
<td>Run 7</td>
<td>200</td>
</tr>
<tr>
<td>Run 8</td>
<td>200</td>
</tr>
<tr>
<td>Run 9</td>
<td>100</td>
</tr>
<tr>
<td>Run 10</td>
<td>100</td>
</tr>
<tr>
<td>Run 11</td>
<td>100</td>
</tr>
<tr>
<td>Run 12</td>
<td>200</td>
</tr>
<tr>
<td>Run 13</td>
<td>200</td>
</tr>
<tr>
<td>Run 14</td>
<td>200</td>
</tr>
<tr>
<td>Run 15</td>
<td>100</td>
</tr>
<tr>
<td>Run 16</td>
<td>200</td>
</tr>
</tbody>
</table>

3.4. Application of the proposed parameter tuning method in the case study problem

3.4.1. Full factorial design of experiment to investigate the significant factors

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

First, the first response variable is investigated. Among all factors and their interactions, only factor A, B, C, D, and the AB interaction are obtained to be significant using full factorial design of experiment. The regression model for R1 is presented as: \( R1 = 753 + 0.480 \times A + 1.110 \times B + 34.6 \times C + 70.4 \times D - 0.01023 \times A \times B \). The main and interaction plots are presented as Figure 1 (a and b). According to Figure 1 (a), all main factors are significant; while Figure 1 (b) represents that only the interaction between factor A and B is significant.

As it is presented in Figure 2, investigating the effect of factors on the second response variable reveals that only factor A, C, D, and AC are significant. The regression model for R2 which only includes the significant factor is obtained as: \( R2 = 131.2 - 0.552 \times A - 177.4 \times C + 158.1 \times D + 0.820 \times A \times C \). The main effect and interaction plots for the second response variable are presented in Figure 2. It should be mentioned that for the sake of simplicity, only significant factors and interactions are considered in creating plots in Figure 2.
According to Figure 3, considering the third response variable as the response variable in the full factorial design of experiment shows that only main factors are significant. In this case, the regression model for R3 only includes factors A, B, C, and D. The regression model for R3 is presented as: \( y = 0.410 + 0.000516 A + 0.000187 B - 0.0538 C - 0.0184 D \). Since there is no significant interactions, only main factors are significant which are presented in Figure 3.

It should be mentioned that in all investigations, the assumption of full factorial design of experiment are tested and the results show no violation of the assumptions.

The summary of result obtained from the full factorial design of experiment is presented in Table 4.

### Table 4: Summary of FFT results considering response variable individually

<table>
<thead>
<tr>
<th>Responses</th>
<th>Factors</th>
<th>Significant factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>A: 200</td>
<td>A, B, C, D, A*B</td>
</tr>
<tr>
<td></td>
<td>B: -</td>
<td></td>
</tr>
<tr>
<td></td>
<td>C: 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>D: 1</td>
<td></td>
</tr>
<tr>
<td>R2</td>
<td>A: 100</td>
<td>A, C, D, A*C</td>
</tr>
<tr>
<td></td>
<td>B: -</td>
<td></td>
</tr>
<tr>
<td></td>
<td>C: 2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>D: 1</td>
<td></td>
</tr>
<tr>
<td>R3</td>
<td>A: 100</td>
<td>A, B, C, D</td>
</tr>
<tr>
<td></td>
<td>B: 100</td>
<td></td>
</tr>
<tr>
<td></td>
<td>C: 2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>D: 2</td>
<td></td>
</tr>
</tbody>
</table>

#### 3.4.2. Desirability function approach to find the optimal setting of the factors

Considering the regression model obtained from full factorial design of experiment as the objective function, the individual and composite desirability function approaches are used to find the optimal setting of each factor. First, individual desirability function approach is used and each response variable is optimized individually. The result are presented in Figure 4, in which the optimal setting of the factors are presented considering the optimal response variables. The summary of results obtained from individual desirability function approach is presented in Table 5.

[Figure 3: The main effect plot for the third response variable]

[Table 4: Summary of FFT results considering response variable individually]

[Figure 4(a): considering first response variable as objective function]
4(b): considering second response variable as objective function

4(c): considering third response variable as objective function

Figure 4: Individual desirability function results for optimizing each response

Table 5: Results of individual desirability function

<table>
<thead>
<tr>
<th>Response</th>
<th>Optimal Solution</th>
<th>Predicted Response</th>
<th>desirability value</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>200 200 1 1</td>
<td>724.054</td>
<td>0.3052</td>
</tr>
<tr>
<td>R2</td>
<td>100 - 2 1</td>
<td>43.4131</td>
<td>0.8695</td>
</tr>
<tr>
<td>R3</td>
<td>100 100 2 2</td>
<td>0.33623</td>
<td>0.3578</td>
</tr>
</tbody>
</table>

After considering each response variable as an objective function individually, all response variables are optimized using desirability function approach, when three response variables are considered as objective functions simultaneously. Results are presented in Figure 5, in which the optimal settings of the factors are presented while all response variables are optimized simultaneously.

As it is presented in Tables 4, 5, and 6, comparing the results obtained from the full factorial design of experiment, individual desirability function approach, and composite desirability function approach shows that although optimizing each response variable individually will provide the better result for each response variable, but the optimal parameter setting would be different when each response variable is optimized individually. For example, considering the first response variable as objective function the optimal setting for factors A, B, C, and D are obtained as 200, 200, 1, and 1, respectively; but when response variable 3 is considered as a response variable, the optimal settings are different. Considering all response variable as objective functions simultaneously in the composite desirability function method, will generate one general setting for all parameters of the algorithm and will lead to an optimal value of all response variables.
The summary of result obtained from the composite desirability function approach is presented in Table 6.

Table 6: Results of composite desirability function

<table>
<thead>
<tr>
<th>Factors</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>Predicted Response</th>
<th>Desirability value</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>100</td>
<td>100</td>
<td>2</td>
<td>1</td>
<td>872.982</td>
<td>0.68842</td>
</tr>
<tr>
<td>R2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>43.4131</td>
<td></td>
</tr>
<tr>
<td>R3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.354677</td>
<td>0.354677</td>
</tr>
</tbody>
</table>

4. Conclusion
The paper proposed a methodology to tune the parameters of the multi-objective evolutionary algorithms based on desirability function. To validate the proposed methodology, it is applied on a case study which was a tri-objective single machine scheduling to minimize makespan, total weighted tardiness, and total weighted completion times. Multi-objective particle swarm optimization applied to solve the problem. In order to tune the parameters of the algorithm, the proposed methodology applied. First, a full factorial design of experiment is applied to find the approximate regression model for each response variable. The regression model only includes the significant parameters of the algorithm which affect the performance of the algorithm. Then, using the regression model as objective function, the composite desirability function approach is utilized to find the optimal setting of the parameters of the algorithm, while all response variables, i.e., performance metrics of the algorithm, are optimized. For future research, the proposed method can be applied in different multi-objective optimization problem, when different evolutionary algorithms are applied. In addition, considering weights for each performance metric of the algorithm can be considered as another interesting extension to this research.

References


Taguchi, G. *System of experimental design; engineering methods to optimize quality and minimize costs*, No. 04; QA279, T3., 1987.


Wang, L., Gordon, M. D., and Zhu, J. Regularized least absolute deviations regression and an efficient algorithm for parameter tuning. In *Data Mining, 2006. ICDM’06. Sixth International Conference on* (pp. 690-700). IEEE.


**Biography**

**Mohammad Komaki** is accomplished B.Sc. in Industrial Engineering at Sharif University of Technology, Tehran, Iran (2001–2006) and M.Sc. in Industrial Engineering at Mazandaran University of Science and Technology, Babol, Iran (2007–2010). He is currently studying System Engineering at Case Western Reserve University, Cleveland, USA. His research interests include optimization and multi-criteria decision making.

**Behnam Malakooti** obtained his PhD in 1982 from Purdue University. He has consulted for numerous industries and corporations, including General Electric, Parker Hannifin, and B.F. Goodrich. He has published over 100 papers in technical journals. In his work, systems architectures, space networks, optimization, multiple criteria & intelligent decision making, trait analysis of biological systems, adaptive artificial neural networks, and artificial intelligence theories and techniques are developed and applied to solve a variety of problems. His current research is on design and protocols analysis for NASA space-based networks. Recently Professor Malakooti developed a four-dimensional approach for decision-making process typology and risk analysis. Decision-making typology accurately identifies the four types of decision makers’ behavior: Information processing, creativity, risk, and decisiveness approach. It also provides a basis for developing the next generation of intelligent robots. See [http://car.cwru.edu/decision/](http://car.cwru.edu/decision/) for computerized survey. He has made contributions to manufacturing systems developing computer aided approaches for manufacturing/production design, planning, operations, facility layout, assembly systems, scheduling, MEMS, and machine set-up, tool design, and machinability.