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. Simt 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 approaches 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).

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

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

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

Next, the individual desirability functions can be integrated as overall (composite or aggregated) desirability (D), which can be between 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.

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

More details about the desirability function approach and its applications are presented in (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∈{1,2,…,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 π be sequence of jobs where $\pi(1)$ represents the job in the first position, then the objectives of sequence π can be computed as following.

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

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

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, 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 following.

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

Total weighted tardiness (TWT) of the sequence π is as following.

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

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

3.2 Evolutionary algorithm to solve the case problem

There are sevral MOEAs to solve the described problem in the periviouse section. MOPSO ,NSGA-II,and NSGA-III are the most commenly 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 themulti-objective X-bar control chat 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 MOPSOS 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 metahesuristic 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_{pop} is one parameter that needs to be set. MOPSO is an iterative algorithm, that is, it iterates It_{max} times by performing same operations at each iteration. The location of each particle b at each iteration t of the alogrithm is presented by s_b^t and its velocity is represented by v_b^t. In this paper, the initial position and velocity of each particle $b = 1,2, \dots, N_{pop}$ is generated randomly and then the largest rule is applied on the s_b^t to find the sequnce 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 upto iteration t is presented by $pBest_b^t$ and the best position found by the swarm is presented 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 following.

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

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 $pBest_b^t$ and $gBest^t$, respectively. Note that w_t in Eq. (11), called the inertia weight of the particle b, controls the effect of the particles velocity at the pervious iteration. Shi and Eberhart (1999) and Naka et al. (2001) suggested using 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 as follows.

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

3.3. Performance measurements of the algorithm

In order to evaluate the performace of MOPSO, several metrics can be used to measure variouse 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 closseness 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 ith non-dominated solution, respectively:

$$MID = \frac{1}{n} \sum_{i=1}^{n} \left(\sqrt{f_{1i} + f_{2i} + f_{3i}}\right) \tag{13}$$

The next perfromance metric, spacing, represents the relative distances of consecutive solutions in the Pareto frontier (Akhavan Niaki et al., 2011). Eq. (14) represents this metric where $dist_i = \min_{k \in E \wedge k \neq i} \sum_{m=1}^{M} \left| f_m^i - f_m^k \right|$ and $\overline{dist} = \sum_{i=1}^{|n|} \frac{dist_i}{n}$.

$$sp = \sqrt{\frac{1}{|n|} \sum_{i=1}^{|n|} (dist_i - \overline{dist})^2} \tag{14}$$

Note that If all solutions in the Pareto frontier are equally spread, then the spacing metric wold be equl to zerro.
The next metric, spread, is proposed by Deb et al. (2002) and mesures the spread of the solutions in the Pareto frontier as presented in Eq. (15).

$$\Delta = \frac{\sum_{m=1}^{M} dist_m^e + \sum_{i=1}^{|E|} \left| dist_i - \overline{dist} \right|}{\sum_{m=1}^{M} dist_m^e + |E| \overline{dist}} \tag{15}$$

where $dist_i$ is the distance between neighbour solutions, $\overline{dist}$ is the average distance, $dist_m^e$ is the distance between the extrme solution of the problems and and E corresponding to the mth objective function. When the solutions are ideally distributed, the spread metric will be zero.
In this reseach, the described performance metrics for the evalotionary algorithm are considered as response variables, which need to be optimized. Table 1 summaries the properties of each response variable.

Table 1: The response variables (performance evaluation) for the MOPSO parameter tuning experiment

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

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 2: The factors (MOPSO parameter) for the MOPSO parameter tuning experiment

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

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

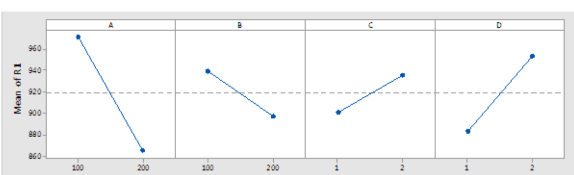
	Factors (algorithm parameters)				Response variables		
	A	B	C	D	R1	R2	R3
Run 1	100	100	1	1	929.6198	130.4420	0.427543
Run 2	200	100	1	1	947.9585	195.2630	0.349748
Run 3	100	200	1	1	870.1583	129.1685	0.408271
Run 4	100	100	2	1	872.9819	0.103935	0.406325
Run 5	100	100	1	2	1042.207	278.8177	0.326788
Run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

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

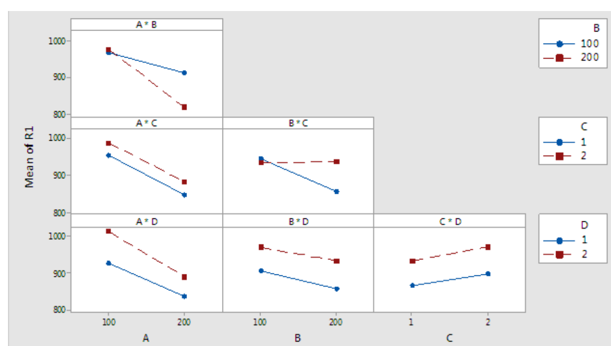
3.4.1. Full factorial design of experiment to investigate the significant factors

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 A + 1.110 B + 34.6 C + 70.4 D - 0.01023 A*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.



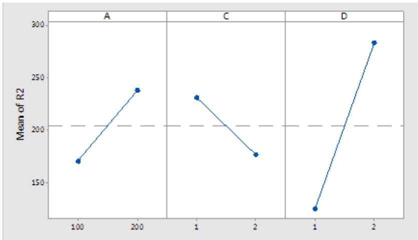
1(a): The main effect plots for R1



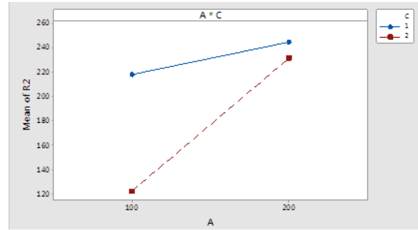
1(b): The interaction plots for R1

Figure 1: The main and interaction plots for the first response variable

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 A - 177.4 C + 158.1 D + 0.820 A*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.



2(a): The main effect plots for R2



2(b): The interaction plots for R2

Figure 2: The main and interaction plots for the second response variable

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: 3 = 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.

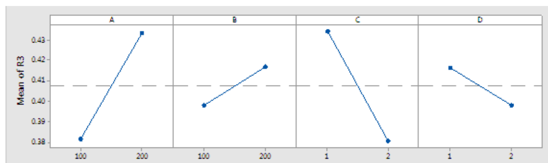


Figure 3: The main effect plot for the third response variable

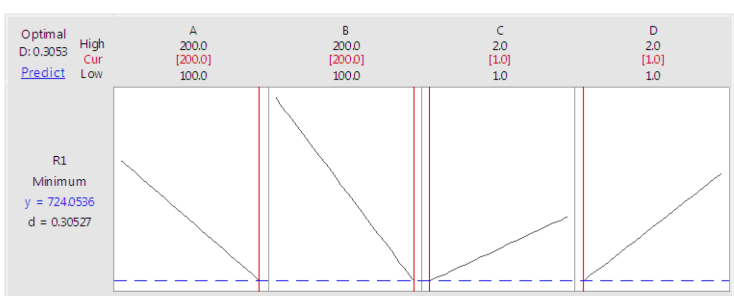
It should be mentioned that in all investigations, the assumption of full factorial design of experiment are tested and the results show no violation of the assumptions.
The summary of result obtained from the full factorial design of experiment is presented in Table 4.

Table 4: Summary of FFT results considering response variable individually

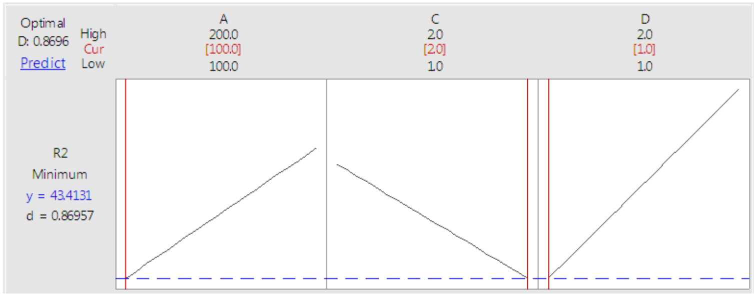
Responses	Factors				Significant factors
	Recommended level				
	A	B	C	D	
R1	200	200	1	1	A, B, C, D, A*B
R2	100	-	2	1	A, C, D, A*C
R3	100	100	2	2	A, B, C, D

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

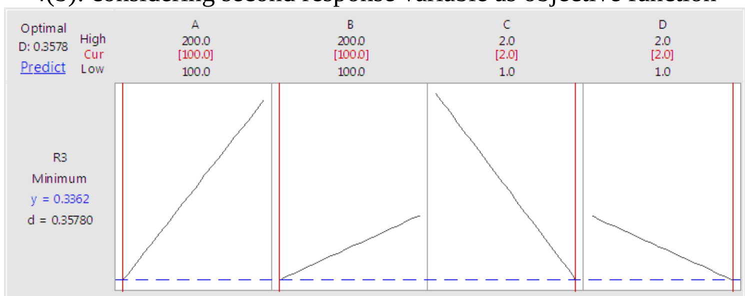
Considering the regression model obtained from full factorial design of experiment as the objective function, the individual and composite desirability function approaches are used to find the optimal setting of each factor. First, individual desirability function approach is used and each response variable 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.



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

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

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.

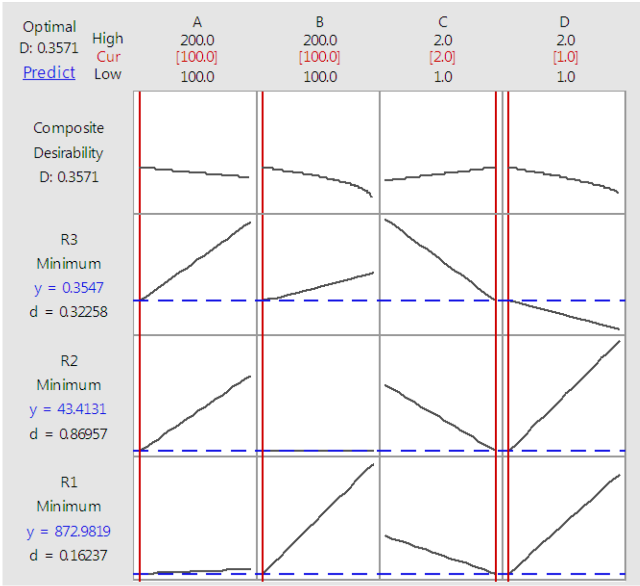


Figure 4: Composite desirability function approach for multi responses optimization results

The summary of result obtained from the composite desirability function approach is presented in Table 6.

Table 6: Results of composite desirability function

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

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

Aboutaleb, A. M., Bian, L., Elwany, A., Shamsaei, N., Thompson, S. M., and Tapia, G. Accelerated Process Optimization for Laser-Based Additive Manufacturing by Leveraging Similar Prior Studies. *IIE Transactions*, 10.1080/0740817X.2016.1189629, 2016.

Akay, Bahriye, and Dervis Karaboga. Parameter tuning for the artificial bee colony algorithm. Computational collective intelligence. Semantic web, social networks and multi-agent systems. Springer Berlin Heidelberg, 608-619. 2009.

Akhavan Niaki, S. T., Malaki, M., and Ershadi, M. J. A comparative study of four evolutionary algorithms for economic and economic-statistical designs of MEWMA control charts, *Journal of Optimization in Industrial Engineering*, no. 9, pp. 1–13, 2011.

Arcuri, A., and Fraser, G. *On parameter tuning in search based software engineering. In Search Based Software Engineering (pp. 33-47).* Springer Berlin Heidelberg. 2011.

Borghei S., Teymourian E., Mobin M., Komaki M., Enhanced imperialist competitive algorithm for the cell formation problem using sequence data, *17th International Conference on Engineering Management*, Chicago, USA, 2015

Coello, C. A. C., Pulido, G. T., and Lechuga, M. S. Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 256- 279, 2004

Crawford, B., Soto, R., Monfroy, E., Palma, W., Castro, C., and Paredes, F. Parameter tuning of a choice-function based hyperheuristic using particle swarm optimization. *Expert Systems with Applications*, vol. 40, no. 5, pp. 1690-1695, 2013.

Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. A fast and elitist multi-objective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

Derringer, G. Simultaneous optimization of several response variables. *Journal of quality technology*, vol. 12, pp. 214-219, 1980.

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

Eiben, Agoston E., and Selmar K. Smit. Parameter tuning for configuring and analyzing evolutionary algorithms. *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 19-31,2011.

Gold, C., Holub, A., and Sollich, P. Bayesian approach to feature selection and parameter tuning for support vector machine classifiers. *Neural Networks*, vol. 18, no. 5, pp. 693-7, 2005.

Goldberg, D.E. *Genetic Algorithms in Search, Optimization, and Machine learning,* first ed., Studies in Computational Intelligence, Addison-Wesley Longman Publishing Co., Inc., 1989.

Graham, R. L., Lawler, E. L., Lenstra, J. K., and Rinnooy Kan, A. H. G., Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annal Discrete Math*, vol. 5, no. 1, pp. 287-326, 1979.

Harrington, E. C., The desirability function. *Industrial quality control*, vol. 21, no. 10, pp. 494-498, 1965.

He, Ben, and Iadh Ounis. A study of parameter tuning for term frequency normalization. *Proceedings of the twelfth international conference on Information and knowledge management. ACM*, 2003.

Imbault, F., and Lebart, K. A stochastic optimization approach for parameter tuning of support vector machines." Pattern Recognition, 2004. ICPR 2004. *Proceedings of the 17th International Conference on*. Vol. 4. IEEE, 2004.

Iwasaki, Nobuhiro, Keiichiro Yasuda, and Genki Ueno. Dynamic parameter tuning of particle swarm optimization. *IEEJ Transactions on electrical and electronic engineering*, vol. 1, no.4, pp. 353-363, 2006.

Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P. Optimization by simulated annealing, *Science*, vol. 220, no. 4598, pp. 671-680, 1983.

Komaki M., Mobin M., Teymourian E., Sheikh M., A general variable neighborhood search algorithm to minimize makespan of the distributed permutation flow shop scheduling problem, *17th International Conference on Engineering Management,* Chicago, USA, 2015.

Kulkarni, Abhijit, Vaidyanathan K. Jayaraman, and Bhaskar D. Kulkarni. Support vector classification with parameter tuning assisted by agent-based technique. *Computers & chemical engineering*, vol. 28, no. 3,pp. 311-318, 2004.

Lawler, E. L., A pseudopolynomial algorithm for sequencing jobs to minimize total tardiness. *Annals of discrete Mathematics*, vol. 1, pp. 331-342, 1977.

Li, Z., Mobin, M, Keyser, T., Multi-objective and multi-stage reliability growth planning in early product development stage, *IEEE Transaction on Reliability*, vol. 65, no. 2, pp. 769-781, 2016.

Mobin, M., Li, Z., and Massahi Khoraskani, M. Multi-objective X-bar control chart design by integrating NSGA-II and data envelopment analysis. *In Proceedings of the 2015 Industrial and Systems Engineering Research Conference*, (pp. 1-10), 2015.

Mobin, M., Mobin, M., and Li, Z. Multi-Response Optimization of Cavitation Peening Parameters for Improving Fatigue Performance Using the Desirability Function Approach. *International Journal of Applied Decision Sciences*, vol. 178, no. 1, pp. 234-240, 2016.

Mobin, M., Roshani, A., Mobin, M., and Rastegari, A. Investigating Cavitation Peening Parameters for Fatigue Performance Using Designed Experiment. *In Proceedings of the 2015 Industrial and Systems Engineering Research Conference* (pp. 1-10), 2015.

Naka, S., Genji, T., Yura, T., and Fukuyama, Y., Practical distribution state estimation using hybrid particle swarm optimization. *In Power Engineering Society Winter Meeting (IEEE)*, 2, 815-820, 2001

Rashidi, E., Parsafard, M., Medal, H., and Li, X. 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. *Transportation Research Part E: Logistics and Transportation Review*, vol. 91, pp. 33-50, 2016.

Salmasnia, A., Moeini, A., Mokhtari, H. and Mohebbi, C., A robust posterior preference decision-making approach to multiple response process design. *International Journal of Applied Decision Sciences*, vol. 6, no. 2, pp.186-207, 2013.

Shi, Y., and Eberhart, R. C. Empirical study of particle swarm optimization. *In Proceedings of the 1999 Congress on Evolutionary Computation* (CEC). 1945–1950, 1999

Smit, Selmar K., and A. E. Eiben. "*Parameter tuning of evolutionary algorithms: Generalist vs. specialist." Applications of evolutionary computation*. Springer Berlin Heidelberg, 2010. 542-551.

Smit, Selmar K., and Agoston E. Eiben. Comparing parameter tuning methods for evolutionary algorithms. Evolutionary Computation, 2009. *CEC'09. IEEE Congress on. IEEE*, 2009.

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

Tavana, M., Li, Z., Mobin, M., Komaki, G., Teymourian, E., Multi-objective control chart design optimization using NSGA-IIIand MOPSO enhanced with DEA and TOPSIS, *Expert Systems With Applications*, vol. 50, pp. 17-39, 2016.

Vafadarnikjoo A., Khatami M., Mobin M., Roshani A., A Meta-Heuristic Approach to Locate Optimal Switch Locations in Cellular Mobile Networks, Proceedings of the 2015 American Society of Engineering Management Conference (ASEM2015), Indiana, USA.

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.

Zitzler, E., Evolutionary algorithms for multi-objective optimization: *Methods and applications*, vol. 63. Citeseer, 1999.

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/ 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.