

RESEARCH

Open Access

Hybrid inner-outer algorithm for solving real-world mechanical optimization problems



Omnia Osman Fadel Abouhabaga^{1*} , Mohamed Hassan Gadallah², Hanan Kamel Kouta¹ and Mohamed Abass Zaghloul¹

* Correspondence: omniaothman@eng.psu.edu.eg

¹Department of Production Engineering & Mechanical Design, Faculty of Engineering, Port Said University, Port Said, Egypt
Full list of author information is available at the end of the article

Abstract

In the real world, the problems mostly are complex; more precisely, the problems generally are nonlinear or large scale other than if it was mandatory to resolve it under certain constraints, and that is common in engineering design problems. Therefore, the complexity of problem plays a critical role in determining the computational time and cost. Accordingly, a novel algorithm called inner-outer array is proposed in this paper. It depends on the design of parameters and then tolerance design as one of design of experiment stages. In this work, the inner-outer algorithm is used to solve real-world optimization problems to choose the preferable feasible regions of the entire search domain. Numerical results are documented and compared based on four well-known constrained mechanical engineering issues. It can be concluded that the performance of inner-outer algorithm is good to optimize constrained engineering problems, but it still needs some enhancements in the future work.

Keywords: Inner-outer array, Constrained optimization, engineering design problems, Hybrid algorithms

Introduction

Computational cost and time of evolutionary algorithms are still the focus of attention of researchers and optimizer without prejudice to the aim of seeking to find the global optima. Evolutionary algorithms have computational costs due to many reasons; the most important of them is the complexity of the problem, like the engineering design problems presented here, computing a large number of fitness functions (multi-objective problems), or difficulty to formulate mathematical model. From here, researchers developed several intelligent algorithms in order to overcome these difficulties, such as genetic algorithm, particle swarm algorithms, and ant colony, as what will be detailed in this section. Tasi et al. [1] proposed a novel approach based on combining the Taguchi method and classical genetic algorithm to solve global optimization problems. The aim of this hybridization was to benefit from the powerful of ability to explore the



© The Author(s). 2021 **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>. The Creative Commons Public Domain Dedication waiver (<http://creativecommons.org/publicdomain/zero/1.0/>) applies to the data made available in this article, unless otherwise stated in a credit line to the data.

global optima and the ability of Taguchi to screen the optimum chromosomes. The proposed approach was applied to 15 benchmark problems in order to investigate the performance of hybrid algorithm and then compare the results, in addition to the approach presented by Vaghela and Prajapati [2], when a hybrid approach of Taguchi and genetic algorithm was proposed to minimize the number of iterations using the powerful of OA to find the initial population for GA. The comparative results showed that HTGA is able to find the optimal solution or close to optimal than the traditional genetic algorithm. Karen et al. [3] proposed a hybrid approach based on the genetic algorithm and Taguchi method to solve a real-world problem in the design and manufacturing field. The hybrid algorithm was applied to vehicle component. The problems related to automotive industry are usually multi-objective problems. The proposed approach has been subjected to verification by evaluating both of fatigue damage and life of two-bar truss. It was found by results that the hybrid algorithm is more effective than classical optimization methods.

Stjepan et al. [4] examined 8 various kinds of crossover operators to comprehend their impact on the performance of the hybrid Taguchi–genetic algorithm. The aim of that investigation was to outline how to use the algorithm in continuous optimization. The different crossover operators have been tested by using 15 benchmark problems as cases study for each operator. It was found that the best crossover operator for successful hybrid Taguchi genetic algorithm performance was segmented besides roulette wheel selection. Wen-Hsien Ho et al. [5] used an intelligent hybrid approach based on the Taguchi method and genetic algorithm in order to minimize each of bearing offset and shaft alignment which occurred in the marine vessels. The hybrid approach was used to reduce the stress and shear force of shaft especially if that shaft preforms under various conditions. Wang and Guo [6] developed a hybrid algorithm to achieve the optimal structure of rotor of IMP machine that used in electric vehicles. The hybrid algorithm was a combination between genetic algorithm and Taguchi method. They concluded that the hybrid Taguchi genetic algorithm had the ability to fix the deficiency of Taguchi method and genetic algorithm for obtaining the global optimum. Zhang et al. [7] presented a comparison study for designing planar thinned arrays, which was used in antenna, based on hybrid orthogonal genetic algorithm (OGA). The results of using OGA had been compared with previous published results. The investigation results showed that OGA reached a better result than those in the previous work. Hegab et al. [8] used DOE to study the effective parameters of electrical discharge machining (EDM) which was used in machining model of AL-CNT. A fractional factorial design L27 for 3-level 5 parameters has been used to indicate which of them is a statistically significant parameter for each response. The studied parameters are machining on time, discharge current, voltage, total depth of cut, and % wt, and the primary objectives are material removal rate, wear electrode ratio, and average surface roughness.

One of the essential features of real-life optimization issues is that they are nonlinear; therefore, it is time-consuming to obtain acceptable solutions for these nonlinear issues. Heuristic methods are hence typically used to speed up the search for the solution. In this light, Alfarisy et al. [9] chose the genetic algorithm (GA) and simulated annealing (SA) among most popular heuristic approaches because of their ability to get a nearly optimal solution in the narrow searching area. They compared the results of

the three types of hybridization of GA and SA applied on nonlinear benchmark functions. They concluded that combining GA with SA enhances the performance of both techniques to solve nonlinear issues. There is a combination between traditional hybrid genetic algorithm (HGA) and local search (LS) method [10] in order to get a simple and economic method. The LS method is based on two methods, which are three-directional search (TD) and Neld-Mead simplex. This hybridization method was proposed to gather between finding global solution with lower cost. This work [11] underlines the importance of hybrid evolutionary algorithms and also presented the several of potentials for hybridization evolutionary algorithms. Considering the preservation of critical features of evolutionary algorithm is the population-based collective learning process, self-adaptation, and robustness. Although the integration between GA and gravitational search algorithm (GSA) [12] has an ability to find the optimal regions of complex search spaces through the interaction of individuals in a population of particles, it suffers from premature convergence particularly when solving complex problems. Therefore, it was hybridized with genetic algorithm to take advantage of its ability to enhance search and avoid premature convergence. One of the newest approaches are proposed under that topic; in the research of Zhange et al. [13], a novel proposal approach integrates a genetic algorithm (GA), an artificial neural network (ANN), multivariate regression analysis (MRA), and a fuzzy logic controller (FLC) to improve the inside environment and energy consumption of building design. One of the critical engineering applications which benefited from hybridization in optimization field is aerospace and automobile manufacturing. As it is known, aluminum is the main element in industrialization of aerospace and automobile due to its lightweight and high resistance to corrosion. Despite its benefits, it is complicated to cut by laser process because its thermal conductivity and reflection are high. A hybrid approach was presented [14] to overcome this problem; this approach depends on combining artificial neural network (ANN)–particle swarm optimization (PSO) algorithm and artificial neural network (ANN)–genetic algorithm (GA). The results had a good indicator of optimizing the parameters for minimum roughness, notch width, and notch taper. Another used algorithm for solving commonly constrained optimization problems is the Tree-Seed algorithm, with several modifications [15–17] added and then compared to other algorithms after implementing them on the same benchmark problems which are often real-life problems like a transmission compressor design, electrical motor, and others.

In the real world, the problems mostly are complex; more precisely, the problems generally are nonlinear or large scale other than if it was mandatory to resolve it under certain constraints, and that is common in engineering design problems. Therefore, the complexity of the problem still plays a critical role in determining the computational time and cost. Consequently, the inner-outer array is presented as a promising algorithm as a guide in achieving that goal.

Methods

Design of experiments

Design of experiments (DOE) is a body of knowledge that helps an investigator to perform better experiments, efficiently analyze data, and establish ties between research

findings and original study objectives. Experimental design is defined as a statistical method used to study the effects of various parameters at the same time. Thus, it can be widely used to define the best input combination and reach good conditions for improving product or process quality and solve manufacturing issues. Based on DOE definition, it can be concluded that the parameters play an effective role in influencing the outcome according to their nature.

Hybrid inner-outer approach

As mentioned previously, the proposed hybrid inner-outer algorithm for solving non-linear bound-constrained optimization problems depends on principles of orthogonal arrays and design of experiments approaches. The aim of this algorithm is to narrow down the search domain of the objective functions and accordingly reducing computational time and cost.

Inner-outer array formula

The developed algorithm is called inner-outer array which depends on two stages of design of experiments viz design of parameters and then tolerance design. The layout of this algorithm consists of two or more arrays depending on the number of factors to be studied in the experiment. The algorithm consists of one inner array and one or more outer arrays which depend on the factor number; each array differs from the other in the number of runs that are determined based on the number of factors assigned in this array and the number of levels for each factor. If there are k factors with n -level for each factor, they can be separated to inner and outer arrays. The proposed inner-outer algorithm can be expressed by this formula:

$$\text{Inner-Outer Array size} = I^n - O^m$$

I: refers to inner arrays.

O: refers to outer arrays.

n : refers to number of inner arrays.

m : refers to number of outer arrays.

Inner-outer array algorithm

Following the algorithm, it illustrates the step sequence of evaluating fitness functions using the inner-outer algorithm. The steps involved in *I - O approach* are shown in the inner-outer array algorithm:

Step 1: Identify the problem.

Step 2: State Objectives $F(x) = F_1(x), F_2(x), \dots, F_n(x)$.

Step 3: Define Output Response(s) $F_i(x)$ and the goal for $F_i(x)$.

$$\min_x F_i(x)$$

$$\max_x F_i(x)$$

Step 4: Identify input Variables $x = x_1, x_2, \dots, x_n$

(i) State sensitive variables (S. Var.)

(ii) State low sensitive or insensitive variables (Ins. Var.)

Step 5: Define Variable Levels

Step 6: Design Experiments

- (i) Select a proper Inner Array (**I-A**) for Ins. Var.
- (ii) Select a proper Outer Array(s) (**O-A**) for S. Var.

Step 7: Conduct experiment and collect data.

Step 8: Interpret results.

Step 9: Do the following until the termination condition is met:

- (i) Increase variable levels by unity.
- (ii) Increase size or number of **O-A**

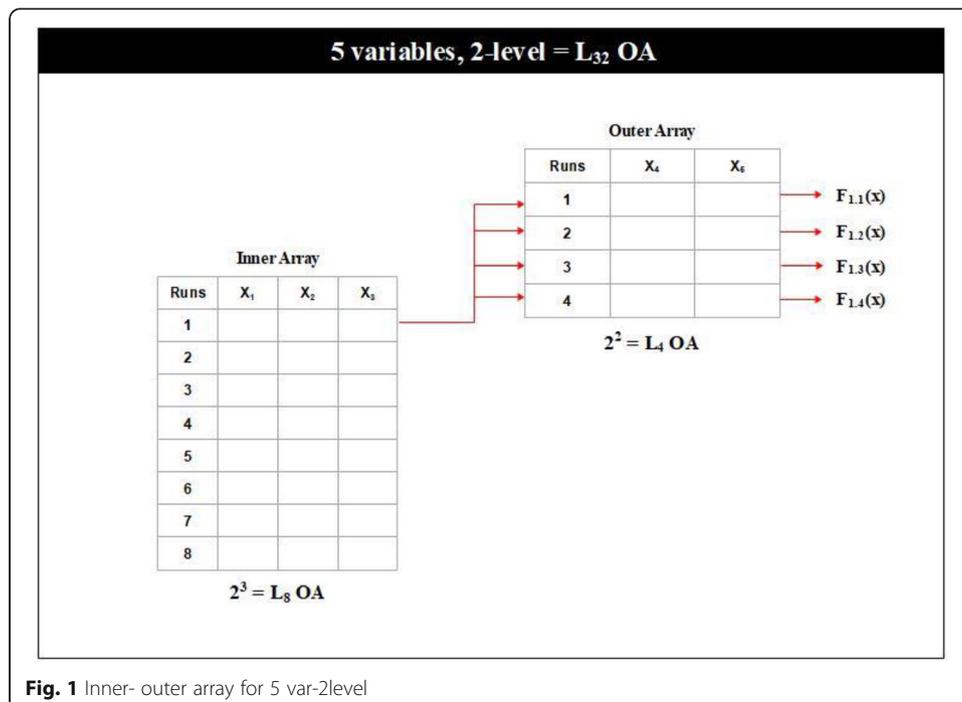
$$F_i(\mathbf{x})$$

(Evaluate)

- (iii) Find the best condition of the trials and compare with earlier best condition. Sort the best one as the best condition.

Step-10: Print the best condition.

Step-11: Stop.



Procedure of inner-outer algorithm

For more clarification, Fig. 1 shows how to arrange 5 variables ($x_1, x_2, \dots,$ and x_5); each variable has 2 levels, with each other in two arrays and then the procedure of evaluating the objective function $F(x)$.

The size of the inner-outer array depends on the number of variables, whose effect on the response will be studied, whereas the number of variables, no matter how many, do not represent an obstacle for this algorithm; as it shown in Fig. 2, there are 7 variables, and 9 variables can be arranged on inner-outer tables according to the importance of these variables.

The inner-outer array is designed to treat simply with the significant variables and try to control them and their critical influence on the design by assigning them in outer arrays, as if focusing on them more than others without using of large-scale arrays at the same time.

Application of the proposed in-out algorithm to real-life engineering problems

Optimization engineering design issues are usually subjected to certain constraints. The main objective of constrained optimization problem is to find the best value of x that satisfies the objective function.

$$\text{minimize } f(\mathbf{x}), \mathbf{x} = (x_1, x_2, \dots, x_i, \dots, x_n)$$

Subject to:

$$g_j(\mathbf{x}) \leq 0, \text{ for } j = 1, \dots, q \text{ (inequality constraint)}$$

$$h_j(\mathbf{x}) = 0, \text{ for } j = q + 1, \dots, m \text{ (equality constraint)}$$

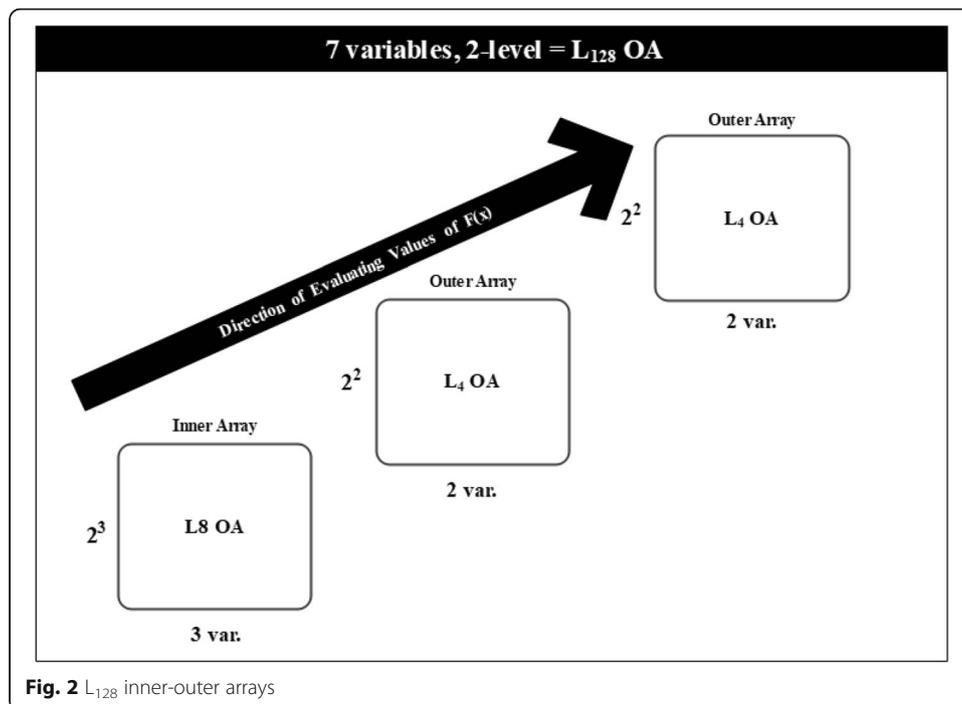


Fig. 2 L_{128} inner-outer arrays

x_i represents the design parameter, and x is the problem solution, while $f(x)$ is the fitness function which needs to be minimized, and m is the number of problem constraints. To demonstrate the competence and robustness of the proposed algorithm, four well-known constrained optimization engineering problems, pressure vessel, speed reducer, gear train, and disc brake, are exhibited in this section.

Pressure vessel design problem

In this work, pressure vessel is one of the applications that are used to investigate the proposed algorithms and also used in several works before [18, 19]. Pressure vessel issue is studied to reduce the material cost of each shell, forming, and welding. The objective function of pressure vessel is a nonlinear and is subject to one nonlinear inequality constraint, and the three others are linear. To be precise, there are four design variables, as shown in Fig. 3, associated with it. Structure parameters of pressure vessel problem, thickness of the pressure vessel (T_s), thickness of the head (T_h), inner radius of the vessel (R), and length of the vessel without heads (L) correspond to x_1 , x_2 , x_3 , and x_4 respectively.

This issue can be mathematically modeled as follows:

$$\text{Min } f(x) = 0.6224 x_1 x_3 x_4 + 1.7781 x_2 x_3^2 + 3.1661 x_1^2 x_4 + 19.84 x_1^2 x_3$$

Subject to

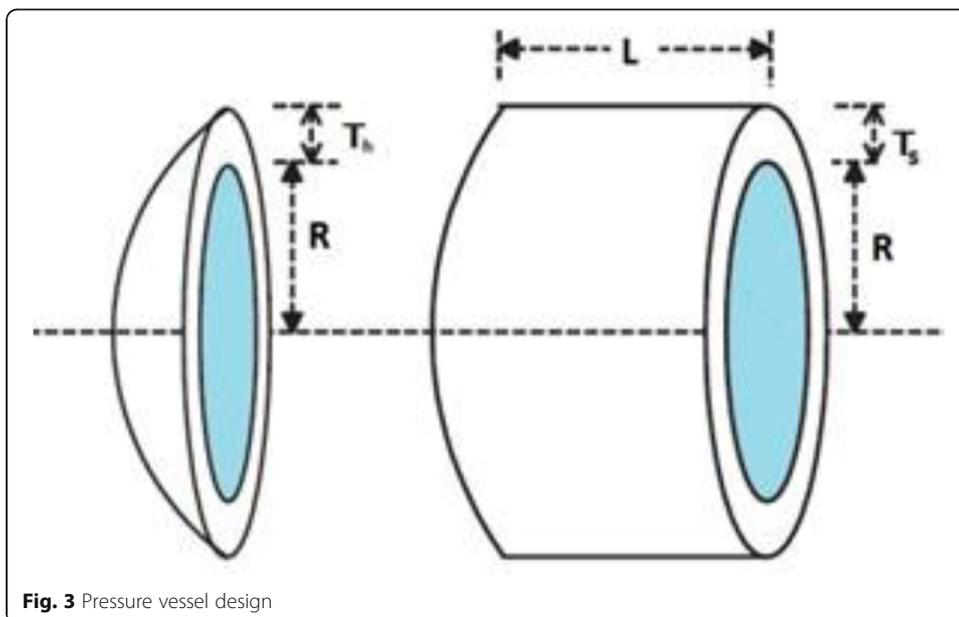
$$g_1(x) = -x_1 + 0.0193 x_3 \leq 0$$

$$g_2(x) = -x_2 + 0.0095 x_3 \leq 0$$

$$g_3(x) = -\pi x_3^2 x_4 - \frac{4}{3} \pi x_3^3 + 1296000 \leq 0$$

$$g_4(x) = x_4 - 240 \leq 0$$

where



$$1 \times 0.0625 \leq x_1, x_2 \leq 99 \times 0.0625$$

$$10 \leq x_3, x_4 \leq 200$$

Speed reducer design problem

The presented speed gear reducer design seeks to minimize its weight according to some constraints related to bending stress of the gear teeth, surface stress, transverse deflections of the shafts, and stresses in the shafts. This is used as an investigate optimization model in several research [18, 20]. There are four linear and seven nonlinear inequality constraints for this issue. There are seven design parameters related with gear reducer to be specific the face width $x_1 = b$, module of teeth $m = x_2$, number of teeth in the pinion $z_1 = x_3$, length of input shaft between bearings $l_1 = x_4$, length of output shaft between bearings $l_2 = x_5$, diameter of input shaft $d_1 = x_6$, and diameter of output shaft $d_2 = x_7$. They are illustrated in the next schematic diagram, Fig. 4.

This optimization problem can be stated as follows:

$$\begin{aligned} \text{Min } f(x) = & 0.7854 x_1 x_2^2 (3.3333 x_3^2 + 14.9334 x_3 - 43.0934) \\ & - 1.508 x_1 (x_6^2 + x_7^2) + 7.4777 (x_6^3 + x_7^3) \\ & + 0.7854 (x_4 x_6^2 + x_5 x_7^2) \end{aligned}$$

Subject to

$$g_1(x) = \frac{27}{x_1 x_2^2 x_3} - 1 \leq 0$$

$$g_2(x) = \frac{397.5}{x_1 x_2^2 x_3^2} - 1 \leq 0$$

$$g_3(x) = \frac{1.93 x_4^3}{x_2 x_3 x_6^4} - 1 \leq 0$$

$$g_4(x) = \frac{1.93 x_5^3}{x_2 x_3 x_7^4} - 1 \leq 0$$

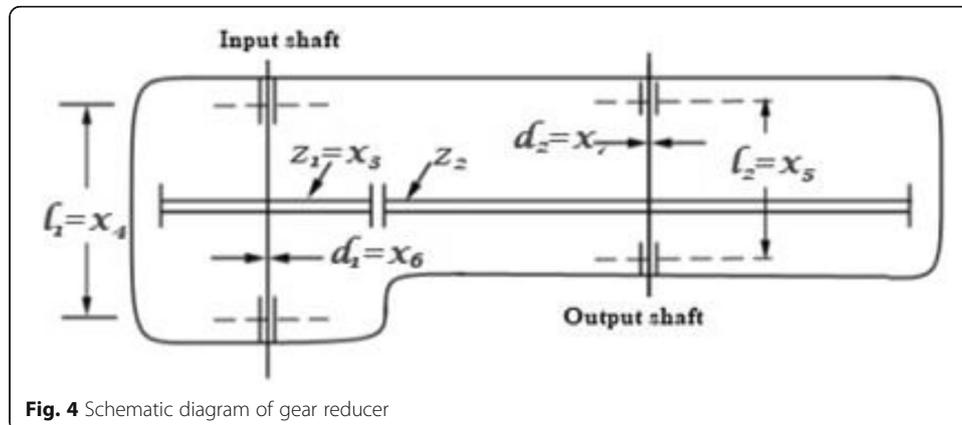


Fig. 4 Schematic diagram of gear reducer

$$g_5(x) = \frac{\left(\left(\frac{745 x_4}{x_2 x_3}\right)^2 + 16.9 \times 10^6\right)^{1/2}}{110 x_6^3} - 1 \leq 0$$

$$g_6(x) = \frac{\left(\left(\frac{745 x_4}{x_2 x_3}\right)^2 + 157.5 \times 10^6\right)^{1/2}}{85 x_7^3} - 1 \leq 0$$

$$g_7(x) = \frac{x_2 x_3}{40} - 1 \leq 0$$

$$g_8(x) = \frac{5 x_2}{x_1} - 1 \leq 0$$

$$g_9(x) = \frac{x_1}{12 x_2} - 1 \leq 0$$

$$g_{10}(x) = \frac{1.5 x_6 + 1.9}{x_4} - 1 \leq 0$$

$$g_{11}(x) = \frac{1.1 x_7 + 1.9}{x_5} - 1 \leq 0$$

where

$$2.6 \leq x_1 \leq 3.6; 0.7 \leq x_2 \leq 0.8; 17 \leq x_3 \leq 28; 7.3 \leq x_4 \leq 8.3; 7.8 \leq x_5 \leq 8.3; 2.9 \leq x_6 \leq 3.9; 5 \leq x_7 \leq 5.5$$

Gear train problem

Let us consider a gear train design problem here [15, 19], which consists of four gears A, B, C, and D as shown in Fig. 5. The cost of the gear ratio is the aim of this problem as seeks to minimize it.

The gear ratio is modeled as follows:

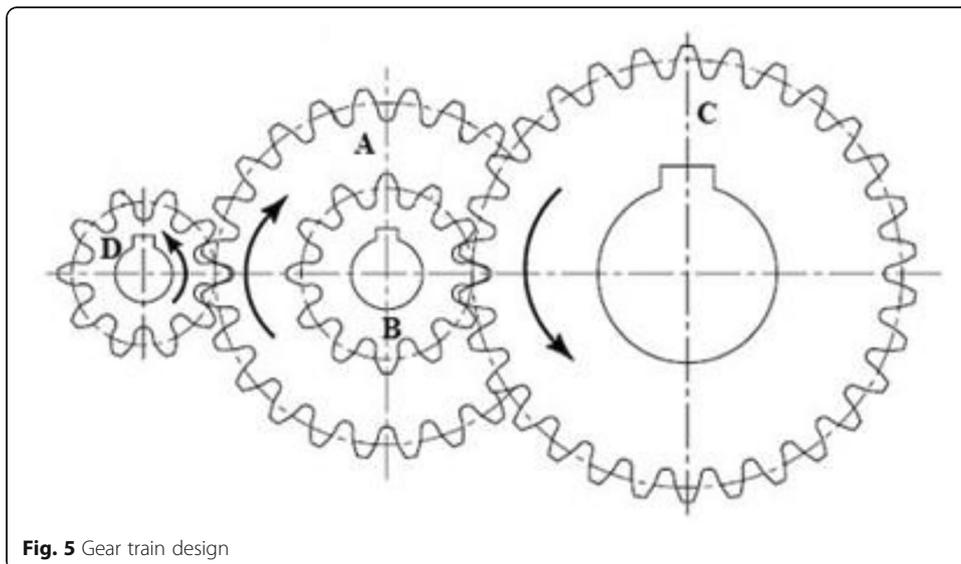


Fig. 5 Gear train design

$$\text{gear ratio} = \frac{n_B n_D}{n_C n_A}$$

This problem is linear without any constraints and has four design variables n_A , n_B , n_C , and n_D are denoted as x_1 , x_2 , x_3 , and x_4 respectively, with boundary condition. These variables have to be integral number of teeth, and also, they are in discrete form. The problem can be mathematically defined as:

$$\text{Min } f(x) = \left(\frac{1}{6.931} - \frac{x_3 x_2}{x_1 x_4} \right)^2$$

where

$$12 \leq x_1, x_2, x_3, x_4 \leq 60$$

Disc brake problem

The goal in this problem is to reduce both the brake mass and the stopping time in the design of a multiple-disc brake [19]. There are six inequality constraints the design of disc brake is subjected to; they are the minimum distance between the radii, the maximum length of the brake, the pressure, the temperature, and the torque limitations. There are four design variables for disc brake design; they are the inner radius of the discs, the outer radius of the discs, the engaging force, and the number of friction surfaces. The formulation of this design can be expressed as follows:

$$\text{Min } f(x) = \frac{(4.9(x_2^2 - x_1^2)(x_4 - 1))}{100000}$$

Subject to

$$g_1(x) = \frac{98 \times 10^4 (x_2^2 - x_1^2)}{x_3 x_4 (x_2^3 - x_1^3)} - 32 \leq 0$$

$$g_2(x) = 20 - (x_2 - x_1) \leq 0$$

$$g_3(x) = 2.5(x_4 + 1) - 30 \leq 0$$

$$g_4(x) = \frac{x_3}{3.14(x_2^2 - x_1^2)} - 0.4 \leq 0$$

$$g_5(x) = \frac{22.2 \times 10^{-4} x_3 (x_2^3 - x_1^3)}{(x_2^2 - x_1^2)^2} - 1 \leq 0$$

$$g_6(x) = 900 - \frac{266 \times 10^{-4} x_3 x_4 (x_2^3 - x_1^3)}{(x_2^2 - x_1^2)} \leq 0$$

where

$$55 \leq x_1 \leq 80; 75 \leq x_2 \leq 110;$$

$$1000 \leq x_3 \leq 3000; 2 \leq x_4 \leq 20$$

Verification of proposed algorithm

There are four viewpoints have been used to validate the performance of the proposed algorithm through them. Each one of them has been discussed separately and then displays the results that have been accessed.

Studying a local optimum solution

After applying the orthogonal arrays to the case of the study and obtaining the results, the first result following the global optimum value was studied by focusing on it so that a larger size of the matrix was used. In other words, the tolerance design procedure was applied to it and the number of levels increased and then the extent of its impact on the results was studied.

For example, if L4 OA is used to express 2-level, then the number of levels can be increased to 4 and hence L8 OA is used. Table 1 shows the original inner-outer array to pressure vessel design problem, where the optimal value 86236.7 was achieved when values of x_1 , x_2 , x_3 , and x_4 are 6.1875, 0.0625, 105, and 10, respectively.

The first result after that optimal was 136166.7735 at variables; values are $x_{1,...,5} = \{6.1875, 0.0625, 105, 105\}$, and Table 2 illustrates the implementation of tolerance design (upsized levels) at this condition.

It is clear that the result is improved by studying the next local value of the optimal and applying the tolerance design to it.

Inner and outer arrays swapping

The results of the inner-outer array are investigated by swapping between both the inner matrix and the outer matrix. To clarify, if the inner array is L4 OA, where it expresses two variables with 2-level, and the outer array is L9 OA for 3-level of two variables, here both the number of levels and experiments are swapped between the inner and outer arrays, so that the size of the inner array will equal 9 runs for 3-level and the outer array will be L4 for 2-level, as shown in Tables 3 and 4.

When the inner-outer array is constructed as L4 OA for inner and L9 OA for outer, the best value for the objective function that satisfies all constraints is 86236.660 (before swapping). The value of best solution became 165275.1 after swapping the size of arrays, as it became L9 OA for inner array and L4 OA for the outer.

Table 1 Inner-outer arrays of pressure vessel design

Inner Array $3^2=L9$ OA			Outer Array $2^2=L4$ OA			Obj. Fcn
Run	x_3	x_4	Run	x_1	x_2	F(x)
1	10	10	1	0.0625	0.0625	1274.328207
2	10	105	2	0.0625	6.1875	121346.0873
3	10	200	3	6.1875	0.0625	86236.66086
4	105	10	4	6.1875	6.1875	206308.4199
5	105	105				
6	105	200				
7	200	10				
8	200	105				
9	200	200				

Table 2 Inner-outer array of next local optimum condition

Inner Array 3 ² = L9 OA			Outer Array 4 ² = L16 OA			Obj. Fcn F(x)
Run	x3	x4	Run	x1	x2	
1	99.75	99.75	1	5.259375	0.0625	
2	99.75	105	2	5.259375	0.065625	97154.78
3	99.75	110.25	3	5.259375	0.06875	97210.07
4	105	99.75	4	5.259375	0.071875	97265.36
5	105	105	5	5.56875	0.0625	97320.65
6	105	110.25	6	5.56875	0.065625	106758.4
7	110.25	99.75	7	5.56875	0.06875	106813.7
8	110.25	105	8	5.56875	0.071875	106869
9	110.25	110.25	9	5.878125	0.0625	106924.2
			10	5.878125	0.065625	116801.3
			11	5.878125	0.06875	116856.6
			12	5.878125	0.071875	116911.9
			13	6.1875	0.0625	116967.1
			14	6.1875	0.065625	127283.5
			15	6.1875	0.06875	127338.8
			16	6.1875	0.071875	127394
						127449.3

Mixed inner-outer array

In inner-outer algorithm is possible to increase the number of outer arrays based on the number of variables, and here, this algorithm overcomes the problem of using mixed arrays that cannot be subjected to fractionation, especially in the case of many variables with different levels, for example, if there are six variables with three levels and four others with two levels, in this case, the size of the array is 3⁶ × 2⁴ = 11664, and this is very huge size.

Speed reducer design problem has seven variables at different levels; there are five variables with 2-level and two factors with 3-level in normal; it is represented by mixed array which its size is 2⁵ × 3² equal 288 runs. This size has been reduced by almost half by using the inner-outer array to be only 144 runs, as illustrated in Table 5.

Table 3 Before swapping

Inner Array 2 ² =L4 OA			Outer Array 3 ² = L9 OA		
Run	x1	x2	Run	x3	x4
1	0.0625	0.0625	1	10	10
2	0.0625	6.1875	2	10	105
3	6.1875	0.0625	3	10	200
4	6.1875	6.1875	4	105	10
			5	105	105
			6	105	200
			7	200	10
			8	200	105
			9	200	200

Table 4 After swapping

Inner Array 3 ² = L9 OA			Outer Array 2 ² =L4 OA		
Run	x1	x2	Run	x3	x4
1	0.0625	0.0625	1	10	10
2	0.0625	3.125	2	10	200
3	0.0625	6.1875	3	200	10
4	3.125	0.0625	4	200	200
5	3.125	3.125			
6	3.125	6.1875			
7	6.1875	0.0625			
8	6.1875	3.125			
9	6.1875	6.1875			

Convergence of results

First, the relationship between the objective function and the number of iterations to study convergence is represented; secondly is applying ANOVA to find out which of the problem variables are the most and least significant. Subsequently, the least significant variable is fixed at a certain level and then the relationship between the objective function and the number of iterations is plotted again on the same curve to find out whether the convergence started from the same point or not.

Figures 6 and 7 show the relation between the number of iterations and objective function values before and after ANOVA implementation for both issues speed reducer design and disc brake design. Unfortunately, the convergence is not clear on these curves that may be occurred because the selective levels are unsuitable to represent these problems.

Figure 8 shows the relation between objective function and iterations before and after applying tolerance design stage on gear train design problem. The result shows the convergence is beginning from the first iteration.

Table 5 Inner-outer array of speed reducer design

Inner Array L8 OA						Outer Array L9 OA		
Run	x1	x2	x3	x5	x7	Run	x4	x6
1	2.6	0.7	17	8.3	5.5	1	7.3	2.9
2	2.6	0.7	28	8.3	5	2	7.3	3.4
3	2.6	0.8	17	7.8	5.5	3	7.3	3.9
4	2.6	0.8	28	7.8	5	4	7.8	2.9
5	3.6	0.7	17	7.8	5	5	7.8	3.4
6	3.6	0.7	28	7.8	5.5	6	7.8	3.9
7	3.6	0.8	17	8.3	5	7	8.3	2.9
8	3.6	0.8	28	8.3	5.5	8	8.3	3.4
						9	8.3	3.9

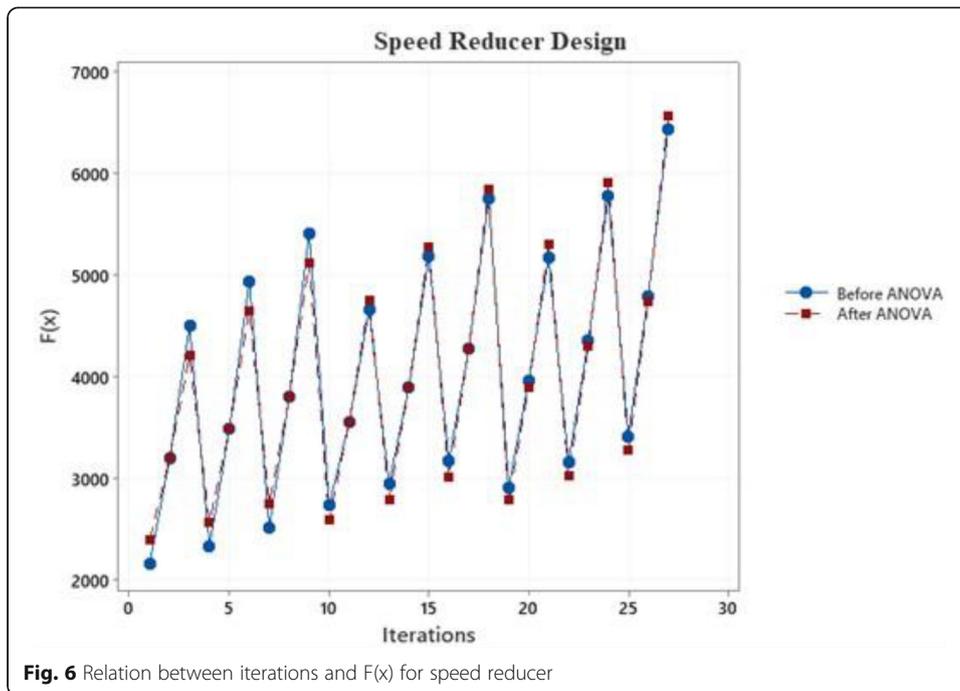


Fig. 6 Relation between iterations and $F(x)$ for speed reducer

Result

The inner-outer algorithm and ANOVA for the stated optimization engineering problems was implemented by using Minitab 19. The suitability of the in-out array was reconfirmed by applying it to solve mechanical optimization engineering problems. A comparison of our results is provided in Table 6 with those obtained by TSA [15], GA-PSO-SQP [20], CI [18], GAMS [21], and ABC [21] algorithms. It is clearly shown that the suggested algorithm leads to acceptable results.

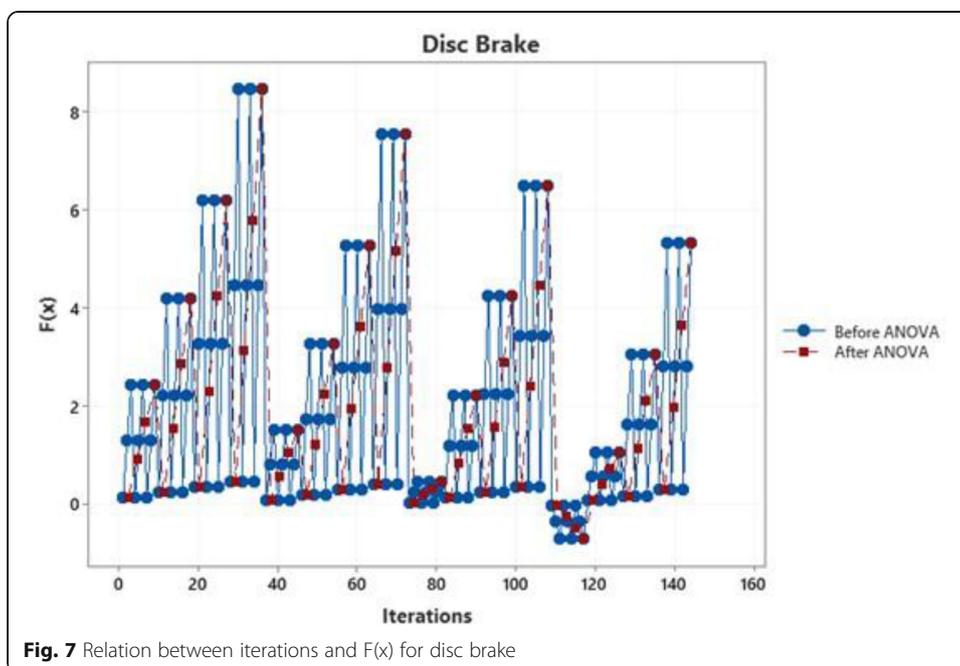
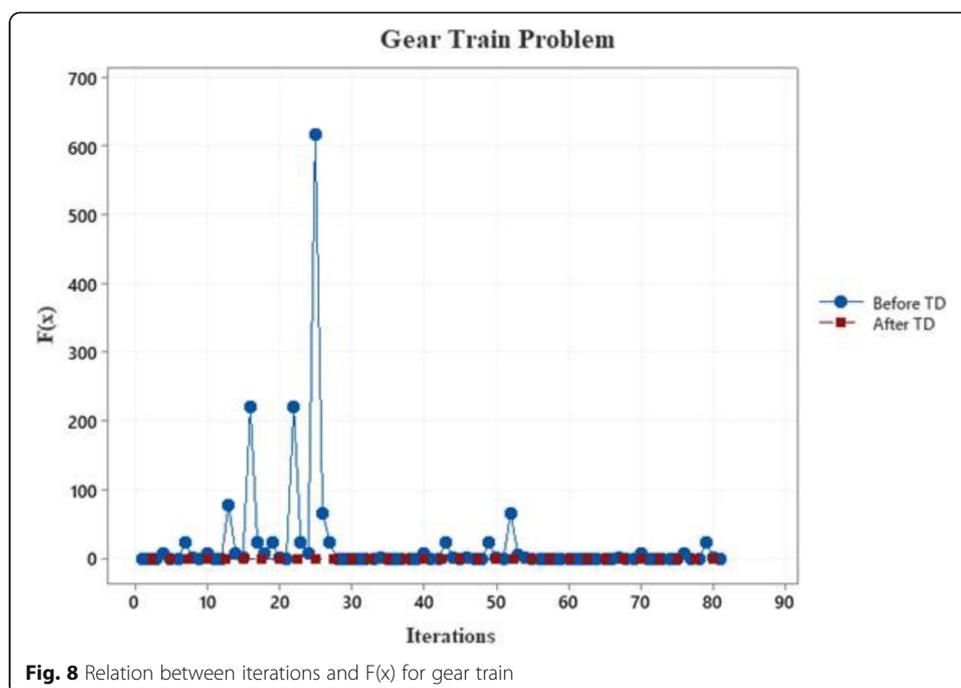


Fig. 7 Relation between iterations and $F(x)$ for disc brake



Discussion

The result of the inner-outer algorithm in designing speed reducer is better with 0.11% than that previously published using hybrid genetic algorithm- particle swarm optimization-sequential quadratic programming (GA-PSO-SQP). In both designs of gear train and disc brake, the results are almost equal to the previously reported results using optimization techniques like artificial bee colony algorithm (ABC) and general algebraic modeling system (GAMS). The outcome of the pressure vessel issue did not achieve the optimum result by using the inner-outer array, but it is clear that the result improved after applying the tolerance design by 5%, indicating that raising the tolerance design percentage improves the result, which will be investigated further.

Levels of the variables in the condition that achieved the optimum result from inner-outer array are as follows: second level for $x_1 = 6.1875$, first level for $x_2 = 0.0625$, second level for $x_3 = 105$, and first level for $x_4 = 10$. By studying the local optimum, as a first verification of proposed algorithm, the first condition, after the optimal condition, its variables levels are second level for $x_1 = 6.1875$, first level for $x_2 = 0.0625$, second level for $x_3 = 105$, and third level for $x_4 = 105$. It is clear that the difference between the levels of the fourth variable x_4 is responsible for the lack of access to optimal value of the objective function.

A worse result for pressure vessel design is obtained, in the second verification, when the size of outer array that includes the $\times 3$ and $\times 4$ is reduced from L9 OA to L4 OA. Accordingly, those variables are more sensitive variables and affected this problem than others.

The problem of using orthogonal arrays for representing large-scale problems is the huge size of these arrays like speed reducer design problem. As its size is too large to be implemented using orthogonal arrays, the proposed method has ability to overcome

Table 6 Comparison of the best solution obtained from various studies for optimization problems

	Present work		TSA	GA-PSO-SQP	CI	GAMS	ABC	
	I-O array	(TD) (5%, 4%, 1.875%)						
Speed reducer	X ₁ =	3.6	3.49631	NA	3.5	NA	NA	3.499
	X ₂ =	0.7	0.7		0.7			0.7
	X ₃ =	17	17		17			17
	X ₄ =	7.3	7.3		7.3			7.3
	X ₅ =	8.05	7.79862		7.7153			7.8
	X ₆ =	3.4	3.3311		3.35			3.3502
	X ₇ =	5.5	5.28843		5.286			5.2878
	F(x)=	3195.21	2991.1355		2994.44355			2997.05841
Pressure vessel	X ₁ =	6.1875	5.56875	NA	0.880944	0.81249	NA	0.8125
	X ₂ =	0.0625	0.0625		0.433702	0.4375		0.4375
	X ₃ =	105	99.75		45.634248	42.098446		42.098446
	X ₄ =	10	10		137.24998	176.636		176.63659
	F(x)=	86236.6608	66916.8895		5798.7989	6059.7144		6059.71434
Gear train	X ₁ =	60	57	NG	NA	NA	NA	49
	X ₂ =	12	13	NG				16
	X ₃ =	36	37	NG				19
	X ₄ =	60	58	NG				43
	F(x)=	0.00058949	0.147E-5	3.0389E-12				0
Disc brake design	X ₁ =	55		NA	NA	NA	55	NA
	X ₂ =	75					75	
	X ₃ =	1000					1561.39914	
	X ₄ =	2					2	
	F(x)=	0.1274					0.1274	

this obstacle by reducing the number of experiments by almost half as what was shown in the third verification.

The fourth verification, its result, shows that the convergence has not yet appeared even after applying ANOVA on both speed reducer and disc brake problems. It was formed when the tolerance design phase was applied to the gear train problem. This is due to the fact that, in general, the levels are treated as discrete, unlike the tolerance design stage, which deals with them as if they are continuous, because they are very narrow.

Conclusion

In this work, an effective optimization approach is developed based on the fundamentals of design of experiments (DOE). This proposed approach was compared to other optimization techniques in order to evaluate its performance and robustness, using four real-life engineering design problems. These problems are nonlinear and constrained optimization problems. In addition, the performance of this algorithm was also verified by four different methods, as was explained before. In light of the above result, it can be concluded that inner-outer algorithm is a promising tool in solving constrained

optimization problems than traditional orthogonal arrays. However, more enhancements are needed in the future; the power of this approach lies in its ability to compare to global optimization strategies such as the genetic algorithm and others, and in some cases, it also outperformed them with fewer iterations, as shown by the results. Hence, the strength of the in-out array is notable in the possibility of combining it with any optimizer techniques.

Abbreviations

I-O: Inner-outer array; TD: Tolerance design; ABC: Artificial bee colony algorithm; PSO: Particle swarm optimization algorithm; CI: Cohort intelligence; GA-PSO-SQP: Hybrid genetic algorithm- particle swarm optimization- sequential quadratic programming; TSA: Tree-Seed algorithm; GAMS: General algebraic modeling system

Acknowledgements

This article is made possible with the assistance and encouragement of both the Faculty of Engineering at Port Said University and Cairo University. The authors would like to thank the reviewers for their suggestions that have enhanced the article presentation.

Authors' contributions

MG performed the conceptualization of the research idea, participated in the interpretation of the results, reviewed the edited manuscript, and supervised the whole project. OA carried out the research methodology, developed the computational technique and performed the investigation and validation of the analytical work, and wrote the manuscript draft. HK and MZ took part in the interpretation of the results and revised the manuscript draft. All authors contributed to the discussion of the results and visualization of the final manuscript.

Funding

No specific funding has to be declared for this work.

Availability of data and materials

The data of hybrid optimization algorithms that support the findings of this study are available in "Springer," https://link.springer.com/chapter/10.1007/978-981-13-1610-4_1

Declarations

Competing interests

The authors declare that they have no competing interests.

Author details

¹Department of Production Engineering & Mechanical Design, Faculty of Engineering, Port Said University, Port Said, Egypt. ²Department of Mechanical Design & Production, Faculty of Engineering, Cairo University, Giza, Egypt.

Received: 15 March 2021 Accepted: 20 May 2021

Published online: 28 July 2021

References

1. Tsai JT, Liu TK, Chou JH (2004) Hybrid Taguchi-genetic algorithm for global numerical optimization. *IEEE Transactions on evolutionary computation* 8(4):365–377. <https://doi.org/10.1109/TEVC.2004.826895>
2. Vaghela PA, Prajapati JM (2019) Hybridization of Taguchi and genetic algorithm to minimize iteration for optimization of solution. *MethodsX* 6:230–238. <https://doi.org/10.1016/j.mex.2019.01.002>
3. Karen I, Yildiz AR, Kaya N, Öztürk NURSEL, Oeztuerk FERRUH (2006) Hybrid approach for genetic algorithm and Taguchi's method based design optimization in the automotive industry. *Int J Prod Res* 44(22):4897–4914. <https://doi.org/10.1080/00207540600619932>
4. Picek S, Golub M, Jakobovic D (2012) Influence of the crossover operator in the performance of the hybrid Taguchi GA. In: 2012 IEEE Congress on Evolutionary Computation (pp. 1-8). IEEE
5. Ho WH, Tsai JT, Chou JH, Yue JB (2016) Intelligent hybrid Taguchi-genetic algorithm for multi-criteria optimization of shaft alignment in marine vessels. *IEEE Access* 4:2304–2313. <https://doi.org/10.1109/ACCESS.2016.2569537>
6. Wang A, Guo J (2017) A novel hybrid genetic algorithm for optimal design of IPM machines for electric vehicle. *Open Physics* 15(1):984–991
7. Zhang L, Jiao YC, Chen B, Li H (2012) Orthogonal genetic algorithm for planar thinned array designs. *Int J Antennas Propagation*. 2012. <https://www.hindawi.com/journals/ijap/2012/319037/>.
8. Hegab HA, Gadallah MH, Esawi AK (2015) Modeling and optimization of Electrical Discharge Machining (EDM) using statistical design. *Manufacturing Review* 2:21. <https://doi.org/10.1051/mfreview/2015023>
9. Fatyanosa TN, Sihananto AN, Alfariy GAF, Burhan MS, Mahmudy WF (2017) Hybrid genetic algorithm and simulated annealing for function optimization. *Journal of Information Technology and Computer Science* 1(2):82–97. <https://doi.org/10.25126/jitecs.20161215>
10. Wan W, Birch JB (2013) An improved hybrid genetic algorithm with a new local search procedure. *J Appl Math*. 2013. <https://www.hindawi.com/journals/jam/2013/103591/>.
11. Grosan C, Abraham A (2007) Hybrid evolutionary algorithms: methodologies, architectures, and reviews. In: *Hybrid evolutionary algorithms* (pp. 1-17). Springer, Berlin

12. Zhang A, Sun G, Wang Z, Yao Y (2015) A hybrid genetic algorithm and gravitational search algorithm for global optimization. *Neural Network World* 25(1):53
13. Zhang T, Liu Y, Rao Y, Li X, Zhao Q (2020) Optimal design of building environment with hybrid genetic algorithm, artificial neural network, multivariate regression analysis and fuzzy logic controller. *Building and Environment* 175: 106810. <https://doi.org/10.1016/j.buildenv.2020.106810>
14. Hamdi M, Zaid M (2019) Resource allocation based on hybrid genetic algorithm and particle swarm optimization for D2D multicast communications. *Applied Soft Computing* 83:105605. <https://doi.org/10.1016/j.asoc.2019.105605>
15. Sahman MA, Cinar AC, Saritas I, Yasar A (2019) Tree-seed algorithm in solving real-life optimization problems. In: IOP conference series: materials science and engineering (Vol. 675, No. 1, p. 012030). IOP Publishing
16. Babalik A, Cinar AC, Kiran MS (2018) A modification of tree-seed algorithm using Deb's rules for constrained optimization. *Applied Soft Computing* 63:289–305. <https://doi.org/10.1016/j.asoc.2017.10.013>
17. Mutluer M, Şahman MA, Çunkaş M (2020) Heuristic optimization based on penalty approach for surface permanent magnet synchronous machines. *Arabian Journal for Science and Engineering* 45(8):6751–6767. <https://doi.org/10.1007/s13369-020-04689-y>
18. Shastri AS, Thorat EV, Kulkarni AJ, Jadhav PS (2019) Optimization of constrained engineering design problems using cohort intelligence method. In: Proceedings of the 2nd International Conference on Data Engineering and Communication Technology (pp. 1-11). Springer, Singapore
19. Akay B, Karaboga D (2012) Artificial bee colony algorithm for large-scale problems and engineering design optimization. *Journal of intelligent manufacturing* 23(4):1001–1014. <https://doi.org/10.1007/s10845-010-0393-4>
20. Belkourchia Y, Azrar L, Zeribab ESM (2019) A hybrid optimization algorithm for solving constrained engineering design problems. In: 2019 5th International Conference on Optimization and Applications (ICOA) (pp. 1-7). IEEE
21. Andrei N, Andrei N (2013) Nonlinear optimization applications using the GAMS technology (pp. 178-181). Springer, New York. <https://doi.org/10.1007/978-1-4614-6797-7>

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)
