Hindawi Publishing Corporation
The Scientific World Journal
Volume 2015, Article ID 936106, 9 pages
http://dx.doi.org/10.1155/2015/936106

Research Article

Conceptual Comparison of Population Based Metaheuristics for Engineering Problems

Oluwole Adekanmbi and Paul Green

Department of Finance and Information Management, Durban University of Technology, P.O. Box 101112, Scottsville, Pietermaritzburg 3209, South Africa

Correspondence should be addressed to Oluwole Adekanmbi; adekanmbioluwole@gmail.com

Received 24 October 2014; Revised 28 December 2014; Accepted 30 December 2014

Academic Editor: Fei Kang

Metaheuristic algorithms are well-known optimization tools which have been employed for solving a wide range of optimization problems. Several extensions of differential evolution have been adopted in solving constrained and nonconstrained multiobjective optimization problems, but in this study, the third version of generalized differential evolution (GDE) is used for solving practical engineering problems. GDE3 metaheuristic modifies the selection process of the basic differential evolution and extends DE/rand/1/bin strategy in solving practical applications. The performance of the metaheuristic is investigated through engineering design optimization problems and the results are reported. The comparison of the numerical results with those of other metaheuristic techniques demonstrates the promising performance of the algorithm as a robust optimization tool for practical purposes.

1. Introduction

In structural engineering, most design optimization problems are highly nonlinear consisting of different design variables and complex constraints such as displacements, geometrical configuration, stresses, and load carrying capability. The design variables are normally grouped into two categories, namely, continuous variables and discrete variables. Optimization problems involving continuous and discrete variables generally require problem-specific search techniques [1]. Evolutionary multiobjective optimization techniques are examples of problem-specific search techniques. Several literatures have applied evolutionary multiobjective optimization techniques to solving multiobjective optimization problems to find a set of trade-off optimal solutions. Since most engineering problems involve multiobjective optimization, it is appropriate to apply an evolutionary optimization algorithm to solve them.

In the last two decades, different types of techniques aimed at effectively and efficiently exploring a search space by combining several basic heuristic methods have emerged [2–4]. These techniques currently referred to as "*Metaheuristics*" are used to describe heuristic methods applied to solving different practical problems. *Metaheuristics* can be considered as a global algorithmic framework used in solving several optimization problems with little changes, thereby making the algorithm adaptive to the specific problem [5].

Metaheuristic search techniques, such as simulated annealing (SA) [6], genetic algorithm (GA) [7], evolution strategies (ESs) [8], and particle swarm optimization (PSO) [9], which are generally developed based on natural phenomena have become the popular optimization techniques of recent years due to their capability of finding promising solutions for complicated optimization problems as well as their independence to the derivatives of objective functions.

Furthermore, metaheuristics can handle both discrete and real-valued variables and can be applied to a wide range of optimization problems effectively. Basically, both trajectory and population based metaheuristic approaches aim to locate the global optimum in the solution space through random moves. The key difference between the metaheuristics is in the way they propose the next move in the solution space. This motivates developers of optimization algorithms to find more efficient methodologies for originating robust optimization algorithms. However, sometimes this results in complicated approaches which are difficult to understand and implement. Hence, this study is an attempt to test the simplicity and efficiency methodology of GDE3 metaheuristics in solving engineering optimization purposes. Section 2 describes the GDE3 metaheuristics briefly. Test cases are described and optimization results are discussed in Section 3. Section 4 provides a clear conclusion of the study.

2. Generalized Differential Evolution Metaheuristic

Several extensions of differential evolution [26] exist for solving constrained and nonconstrained multiobjective optimization problems [27, 28]. In comparison to the extension of differential evolution (DE), GDE3 makes differential evolution a suitable algorithm for multiobjective optimization as well as constrained optimization with little changes to the basic differential evolution algorithm. GDE3 extends DE/rand/1/bin strategy which exhibit slow convergence rates and strong exploration properties. GDE3 is a third version of generalized differential evolution modifying the selection process of the basic differential evolution algorithm [29]. The selection process in GDE3 is guided by these three rules:

(i) In a scenario where both the old vector and trial vector are infeasible, the old vector is selected if it dominates the trial vector, but if the trial vector weakly dominates the old vector, then the trial vector is selected.

(ii) Feasible vector is selected in a situation where both feasible and infeasible vectors are generated.

(iii) In a scenario where both the old vector and trial vector are feasible, the old vector is selected if it dominates the trial vector, but if the trial vector weakly dominates the old vector, then the trial vector is selected.

The whole GDE3 is presented in Algorithm 1. Parts that are new compared to previous GDE versions are framed in Algorithm 1. Without these parts, the algorithm is identical to GDE1. GDE3 can be seen as a combination of GDE2 and Pareto Differential Evolution Approach (PDEA). GDE3 is similar to differential evolution for multiobjective optimization (DEMO) except that DEMO does not contain constraint handling nor recede to basic DE in the case of a single objective because DEMO modifies the basic DE and does not consider weak dominance in the selection. Moreover, GDE3 has an improved diversity maintenance compared to DEMO. There are no constraints to be evaluated when $K = 0$ and $M = 1$, and the selection is simply

$$x_{i,G+1} = \begin{cases} u_{i,G}, & \text{if } f(u_{i,G}) \le f(x_{i,G}), \\ x_{i,G}, & \text{otherwise.} \end{cases} \quad (1)$$

This is the same as for the basic DE algorithm. The size of the population does not increase since this requires that $x_{i,G}$ and $u_{i,G}$ do not dominate each other even weakly, but in the case of a single objective, the reverse is the case. GDE3 performs the sorting of the vector by calculating the crowding distance of the vector. The selection process based on crowding distance gives GDE3 an advantage over NSGAII. In the case of comparing feasible, incomparable, and nondominating solutions, both offspring and parent vectors are saved for the population of the next generation [4]. There is no need to remove elements, since the population size does not increase. Hence, GDE3 is identical to basic DE in this case. GDE3 improves the ability to handle multiobjective optimization problems by giving a better distributed set of solutions and are less sensitive to the selection of control parameter values compared to the earlier GDE versions. As a result, this procedure reduces the computational costs of the metaheuristic and improves its efficiency. Readers interested in GDE3 should refer to the texts by [30, 31].

3. Implementation of Engineering Optimization Problems

The metaheuristic optimization was implemented in NETBEAN v7.3; optimization runs were executed on an HP PC with a 2.30 GHz Intel Dual Core processor and 4 GB of RAM memory. Different examples taken from several optimization literatures were used to show the performance of GDE3 metaheuristic. These examples have been previously solved using a variety of other techniques, which is useful to show the validity and effectiveness of the GDE3 metaheuristic. The optimal results were compared with data recently published in literatures. An experiment has been performed to determine the best values of F and CR for better performance in GDE3 metaheuristic. For this purpose, both CR and F are varied from 0.1 to 1 with an increment of 0.1. The simulations were conducted for each value of F with respect to all values of CR. Hence, 100 such simulations were conducted. From the results, it was found that better Pareto optimal front is obtained by GDE3 with $F = 0.5$, $\mathrm{CR} = 0.9$ and the termination condition is set to the 10,000 objective function evaluations.

Example 1 (welded beam design optimization problem). The welded beam problem is designed to minimize the fabrication cost by subjecting it to some constraints such as bending stress (σ), shear stress (τ), end deflection (δ), and buckling load (P_c). The design variables of the optimization problem are the thickness of the beam (b), the thickness of the weld (h), the welded joint length (l), and the beam width (t). Figure 1 shows the welded beam design structure.

The values of l and h must be integer multiples of 0.0065 in. Assuming $x_1 = h$, $x_2 = l$, $x_3 = t$, and $x_4 = b$ as design variables, the optimization problem can be mathematically expressed as follows:

$$\begin{aligned} \text{Minimize} \quad & f(\vec{x}) = (1 + C_1) {x_1}^2 x_2 \\ & \qquad + C_2 x_3 x_4 (14.0 + x_2), \\ \text{Subject to} \quad & g_1(\vec{x}) = \tau(\vec{x}) - \tau_{\max} \le 0, \\ & g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{\max} \le 0, \\ & g_3(\vec{x}) = x_1 - x_4 \le 0, \end{aligned}$$

```
Input: $D, G_{\max}, NP \geq 4, F \in (0, 1+], CR \in [0,1]$, and initial bounds: $x^{(\text{lo})}, x^{(\text{hi})}$
Initialize: $\begin{cases} \forall i \leq NP \wedge \forall j \leq D\text{: } x_{j,i,0} = x_j^{(\text{lo})} + rand_j[0,1] \cdot (x_j^{(\text{hi})} - x_j^{(\text{lo})}), \\ i = \{1,2,\ldots,NP\}, j = \{1,2,\ldots,D\}, G = 0, rand_j[0,1] \in [0,1] \end{cases}$
While $G < G_{max}$
$\forall i \leq NP$
    Mutation and recombine:
    $r_1, r_2, r_3 \in \{1,2,\ldots,NP\}$, randomly selected,
      except mutually different and different from $i$
    $j_{rand} \in \{1,2,\ldots,D\}$, randomly selected from each $i$
    $\forall j \leq D, u_{j,i,G} = \begin{cases} x_{j,r_3,G} + F \cdot (x_{j,r_1,G} - x_{j,r_2,G}) \\ \qquad \text{if } rand_j[0,1] < CR \vee j == j_{rand} \\ x_{j,r_1,G} \end{cases}$
    Select:
    $x_{i,G+1} = \begin{cases} u_{i,G} & \text{if } f(u_{i,G}) \leq f(x_{i,G}) \\ x_{i,G} & \text{otherwise} \end{cases}$
  Set: $n = n + 1$
    $x_{NP+n,G+1} = u_{i,G}$ if $\begin{cases} \forall j\text{: } g_j(u_{i,G}) \leq 0 \\ \wedge \\ x_{i,G+1} == x_{i,G} \\ \wedge \\ x_{i,G} \nprec u_{i,G} \end{cases}$
while $n > 0$
    Select $x \in \rho = \{x_{1,G+1}, x_{2,G+1}, \ldots, x_{NP+n,G+1}\}$;
      $\begin{cases} x \text{ belongs to the last non-dominated set of } \rho \\ \wedge \\ x \text{ is the most crowded in the last non-dominated set} \end{cases}$
    Remove $x$ from $\rho$
    $n = n - 1$
$G = G + 1$
```

Algorithm 1: The GDE3 algorithm [29].

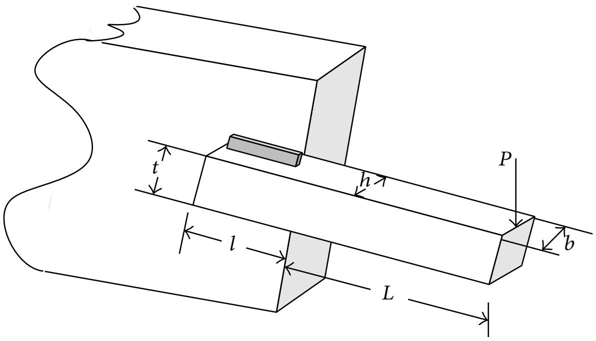


Figure 1: Schematic of the welded beam design problem [1].

$$\begin{aligned} g_4(\vec{x}) &= C_1 \left(x_1{}^2\right) \\ &\quad + C_2 x_3 x_4 \left(14.0 + x_2\right) - 5.0 \leq 0, \\ g_5(\vec{x}) &= 0.125 - x_1 \leq 0, \\ g_6(\vec{x}) &= \delta(\vec{x}) - \delta_{\max} \leq 0, \\ g_7(\vec{x}) &= P - P_c \leq 0, \end{aligned} \tag{2}$$

where

$$\begin{aligned} \tau(\vec{x}) &= \sqrt{(\tau')^2 + (2\tau'\tau'')\frac{x_2}{2R} + (\tau'')^2}, \\ \tau'' &= \frac{MR}{J}, \\ M &= P\left(L + \frac{x_2}{2}\right), \\ R &= \sqrt{\frac{x_2{}^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}, \\ J &= 2\left\{\sqrt{2}x_1 x_2 \left[\frac{x_2{}^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}, \\ \sigma(\vec{x}) &= \frac{6PL}{x_4 x_3{}^2}, \\ \tau' &= \frac{P}{\sqrt{2}x_1 x_2}, \\ \delta(\vec{x}) &= \frac{4PL^3}{E x_3{}^3 x_4}, \\ P_c(\vec{x}) &= \frac{4.013E\sqrt{x_3{}^2 x_4{}^6/36}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right). \end{aligned} \tag{3}$$

TABLE 1: Values of parameters involved in the formulation of the welded beam problem [1].

Constant item	Description	Values
C_1	The welded material	0.10471 ($/in3)
C_2	The bar stock	0.04811 ($/in3)
$\tau_{\max}$	Shear stress of the welded material	13600 (psi)
$\sigma_{\max}$	Normal stress of the bar material	30000 (psi)
$\delta_{\max}$	Bar end deflection	0.25 (inch)
E	Young's modulus of bar stock	30×10^6 (psi)
G	Shear modulus of bar stock	12×10^6 (psi)
P	Loading condition	6000 (lb)
L	Beam's projection length	14 (inch)

The simple bounds of the problem are $x_1, x_4 \in [0.1, 2.0]$ and $x_2, x_3 \in [0.1, 10.0]$. The values of parameters involved in the formulation of the welded beam problem are also shown in Table 1.

The optimum design of the welded beam is executed using GDE3 metaheuristic, and the best solution is found as $x^* = \{x_1, x_2, x_3, x_4\} = \{0.20572840999876, 3.47072911158159, 9.03661683005891, 0.20572540074781\}$ which yields an objective function value of $f(\vec{x}) = 1.7248496$ as seen in Table 2.

The results obtained by GDE3 are presented in Table 2. GDE3 found the global optimum requiring 400 iterations (i.e., 10,000 evaluations) per optimization run. Table 3 provides a comparison of this solution with the results of other optimization algorithms. It is apparent from the table that GDE3 metaheuristic finds a competitive solution using only 10,000 evaluations which is considerably lesser than those of other approaches. Further, a statistical evaluation of 100 independent runs of the GDE3 metaheuristic is tabulated in Table 4 considering the best, worst, average, and the standard deviation (std. dev.) of the obtained solutions. The ratio between the optimized costs corresponding to best and worst designs is 1.00042. Remarkably, GDE3 produced the overall best design result with a value of 1.724849. For continuous optimization problem, [20, 22] found a better design result with a value of 1.7248 at a higher function evaluation.

Example 2 (pressure vessel optimization problem). The pressure vessel problem is designed to minimize total cost which is comprised of the welding cost and forming material cost. The compressed air tank with a working pressure of 3000 psi and a minimum volume of 750 ft^3 must be designed according to the ASME code on boilers and pressure vessels. The design variables of the optimization problem are the length of the cylindrical segment of the vessel (L), the thickness of the cylindrical skin (T_s), the inner radius (R), and the thickness of the spherical head (T_h).

The variables T_s and T_h are discrete values which are integer multiples of 0.0625 inches. Figure 2 shows the cylindrical pressure vessel capped at both ends by hemispherical heads.

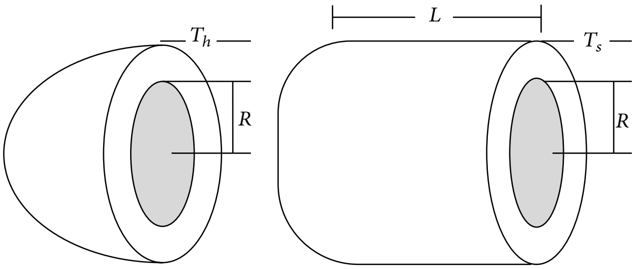


FIGURE 2: Schematic of the pressure vessel design problem [1].

Assuming $x_1 = T_s$, $x_2 = T_h$, $x_3 = R$, and $x_4 = L$ as the design variables, the optimization problem can be mathematically expressed as follows:

$$\begin{aligned}
\text{Minimize} \quad & f(\vec{x}) = 0.6224x_1x_3x_4 + 1.7781x_2{x_3}^2 \\
& \qquad + 3.1611{x_1}^2x_4 + 19.8621{x_1}^2x_3, \\
\text{Subject to} \quad & g_1(\vec{x}) = 0.0193x_3 - x_1 \le 0, \\
& g_2(\vec{x}) = 0.00954x_3 - x_2 \le 0, \\
& g_3(\vec{x}) = x_4 - 240 \le 0, \\
& g_4(\vec{x}) = 750 \times 1728 - \pi {x_3}^2 x_4 \\
& \qquad - \frac{4}{3}\pi {x_3}^3 \le 0.
\end{aligned} \tag{4}$$

The simple bounds of the problem are $x_1, x_2 \in [1 \times 0.0625, 99 \times 0.0625]$ and $x_3, x_4 \in [10.0, 240.0]$. Unlike the usual limit of 200 in considered in literatures, the upper bound of design variable L was increased to 240 in to expand the search space.

Optimization results are presented in Table 5. GDE3 produced a design result with a value of 6083.773 within 400 iterations (i.e., 10,000 evaluations). Table 6 compares the optimal design results produced by GDE3 with those reported in [1, 17, 20, 21, 24, 32]. Further, a statistical evaluation of 100 independent runs of the GDE3 metaheuristic is tabulated in Table 7 considering the best, worst, average, and the standard deviation (std. dev.) of the obtained solutions. The ratio between the optimized costs corresponding to worst and best designs is 1.00229. The best design result was produced by the Firefly algorithm. GDE3 metaheuristic produced the least performance compared to the other algorithms.

Example 3 (speed reducer design optimization problem). The speed reducer design problem [25] is designed to minimize the weight of the speed reducer subjecting it to some constraints such as shaft stresses, surface stress, gear teeth bending stress, and shafts crosswise deflections. The width of the gear face x_1, teeth module x_2, number of pinion teeth x_3, first shaft length between bearings x_4, second shaft length between bearings x_5, the diameter of the first shaft x_6, and diameter of the second shaft are the design variables of the

Table 2: GDE3 solution vector for welded beam.

Best solution	x_1	x_2	x_3	x_4
	0.20572840999876	3.47072911158159	9.03661683005891	0.20572540074781
	$g_1(\vec{x})$	$g_2(\vec{x})$	$g_3(\vec{x})$	$g_4(\vec{x})$
	−0.66062798472194	0.665171394633944	$3.00925094998E - 06$	−3.43299575416113
	$g_5(\vec{x})$	$g_6(\vec{x})$	$g_7(\vec{x})$	$f(\vec{x})$
	−0.08072840999876	−0.235539990649711	0.373971078704926	1.72484969509211

Table 3: Welded beam problem: comparison of GDE3 results with other optimization methods.

Researcher	Metaheuristic	x_1	x_2	x_3	x_4	$f(x)$	NE
[10]	Genetic algorithm	0.2489	6.1730	8.1789	0.2533	2.4331	320,080
[11]	Genetic algorithm	0.2489	6.1097	8.2484	0.2485	2.4000	6,273
[12]	Social behavioral model	0.2407	6.4851	8.2399	0.2497	2.4426	19,259
[13]	Society and civilization algorithm	0.2444	6.2380	8.2886	0.2446	2.3854	33,095
[14]	Genetic algorithm	0.2443	6.2117	8.3015	0.2443	2.3816	320,000
[15]	Particle swarm optimization	0.2444	6.2175	8.2915	0.2444	2.3810	30,000
[16]	Harmonic search	0.2442	6.2231	8.2915	0.2443	2.3810	110,000
[17]	Simulated annealing—direct search	0.2444	6.2158	8.2939	0.2444	2.3811	56,243
[18]	Simulated annealing—genetic algorithm	0.2231	1.5815	12.8468	0.2245	2.2500	26,466
[19]	Artificial Immune System—genetic algorithm	0.2444	6.2183	8.2912	0.2444	2.3812	320,000
[20]	Harmonic search	0.2057	3.4705	9.0366	0.2057	1.7248	200,000
[21]	Simple constrained particle swarm optimizer	0.2057	3.4705	9.0366	0.2057	1.7249	24,000
[22]	Harmonic search—sequential quadratic programming	0.2057	3.4706	9.0368	0.2057	1.7248	90,000
[23]	Differential evolution	0.2444	6.2175	8.2915	0.2444	2.3810	24,000
[8]	Evolutionary algorithm	0.2443	6.2201	8.2940	0.2444	2.3816	28,897
[1]	Firefly algorithm	0.2015	3.5620	9.0414	0.2057	1.7312	50,000
[24]	Simple optimization	0.2057	3.4705	9.0366	0.2057	1.7246	10,000
Present study	Generalized differential evolution 3	**0.2057**	**3.4707**	**9.0366**	**0.2057**	**1.724849**	**10,000**

Table 4: Statistical results of the GDE3 optimization.

Best	Average	Worst	Std. dev.	Number of iterations
1.724849	1.725023	1.725569	0.0001018	400

Table 5: GDE3 Solution vector for pressure vessel.

Best Solution	x_1	x_2	x_3	x_4
	0.74395291436715	0.36774755668330	38.5288195380221	239.37719314082
	$g_1(\vec{x})$	$g_2(\vec{x})$	$g_3(\vec{x})$	$g_4(\vec{x})$
	−0.00034669728332	−0.00018261829057	−0.62280685917400	−42.436889517499
	$f(\vec{x})$			
	6083.77328355025			

Table 6: Pressure vessel problem: comparison of GDE3 results with optimization methods.

Researcher	Metaheuristic	x_1	x_2	x_3	x_4	$f(x)$
[17]	Simulated annealing—direct search	0.7683	0.3797	39.8096	207.2250	5868.76
[32]	Particle swarm optimization—genetic algorithm	0.7500	0.3750	38.8601	221.3654	5850.383
[20]	Harmonic search	0.7500	0.3750	38.8600	221.3600	5849.7
[21]	Simple constrained particle swarm optimizer	0.8125	0.4375	42.0980	176.6360	6.059.714
[1]	Firefly algorithm	0.7500	0.3750	38.8600	221.3600	5850.3
[24]	Simple optimization	1.1250	0.6250	58.2901	43.6927	7199.35
Present study	Generalized differential evolution 3	0.74391	0.36774	38.5288	239.377	6083.773

Table 7: Statistical results of the GDE3 optimization.

Best	Average	Worst	Std. dev.	Number of iterations
6083.773	6092.318	6097.725	40.32205	400

Table 8: GDE3 Solution vector for speed reducer.

Best Solution	x_1	x_2	x_3	x_4
	3.5000000004788	0.7000000000000	17.0000000000000	7.3000000000000
	x_5	x_6	x_7	$g_1(\vec{x})$
	7.8000000000000	3.3502146664526	5.2866832298256	−0.07391528052456
	$g_2(\vec{x})$	$g_3(\vec{x})$	$g_4(\vec{x})$	$g_5(\vec{x})$
	−0.197998527251663	−0.499172248315386	−0.9014716976203	$-3.1892233298E-10$
	$g_6(\vec{x})$	$g_7(\vec{x})$	$g_8(\vec{x})$	$g_9(\vec{x})$
	$-3.8408276559E-11$	−0.7025	$-1.3680001575E-10$	−0.5833333332763
	$g_{10}(\vec{x})$	$g_{11}(\vec{x})$	$f(\vec{x})$	
	−0.0513257534686439	−0.0108523650245949	2996.34816529042	

Table 9: Speed reducer problem: comparison of generalized differential evolution 3 results with simple constrained particle swarm optimization.

Solution	Simple constrained particle swarm optimization [21]	Generalized differential evolution (Present study)
x_1	3.5000	3.5000
x_2	0.7000	0.7000
x_3	17.0000	17.0000
x_4	7.3000	7.3000
x_5	7.8000	7.8000
x_6	3.350214	3.3502146
x_7	5.286683	5.2866832
$f(\vec{x})$	2996.348165	2996.3481653

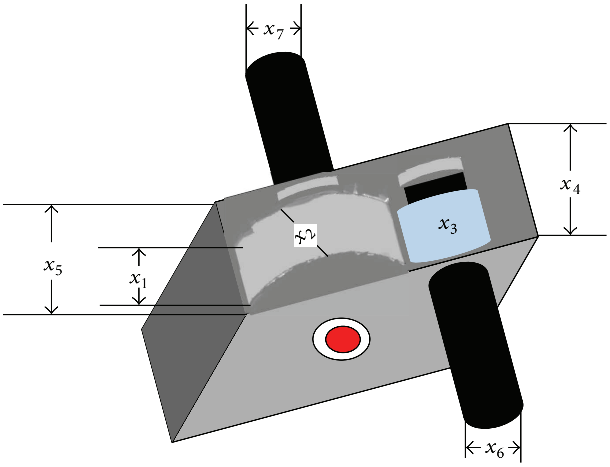


Figure 3: Schematic of the speed reducer design problem [25].

optimization problem. Figure 3 shows the schematic of the speed reducer.

The mathematical expression for the speed reducer problem is as follows:

$$\text{Minimize}\quad f(\vec{x}) = 0.7854x_1x_2^2 \cdot \left(3.3333x_3^2 + 14.9334x_3 - 43.0934\right) - 1.508x_1\left(x_6^2 + x_7^2\right) + 7.4777\left(x_6^3 + x_7^3\right) + 0.7854\left(x_4x_6^2 + x_5x_7^2\right),$$

$$\text{Subject to}\quad g_1(\vec{x}) = \frac{27}{x_1x_2^2x_3} - 1 \le 0,$$

$$g_2(\vec{x}) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \le 0,$$

$$g_3(\vec{x}) = \frac{1.93x_4^3}{x_2x_3x_6^4} - 1 \le 0,$$

$$g_4(\vec{x}) = \frac{1.93x_5^3}{x_2x_3x_7^4} - 1 \le 0,$$

$$g_5(\vec{x}) = \frac{1.0}{110x_6^3} \cdot \sqrt{\left(\frac{745.0x_4}{x_2x_3}\right)^2 + 16.9\times 10^6} - 1 \le 0,$$

$$g_6(\vec{x}) = \frac{1.0}{85x_7^3} \cdot \sqrt{\left(\frac{745.0x_5}{x_2x_3}\right)^2 + 157.5\times 10^6} - 1 \le 0,$$

Table 10: Statistical results of the GDE3 optimization.

Best	Average	Worst	Std. dev.	Number of iterations
2996.3481653	2996.3483815	2996.3491534	0.0000021	400

Table 11: GDE3 Solution vector for tension/compression spring.

Best Solution	x_1	x_2	x_3	$g_1(\vec{x})$
	0.0517955276224998	0.359283196922392	11.1405163630287	$-3.0601282864E-05$
	$g_2(\vec{x})$	$g_3(\vec{x})$	$g_4(\vec{x})$	$f(\vec{x})$
	−0.133636716257444	−4.05865278285946	−0.725947516970072	0.01266583600858

Table 12: Tension/compression spring problem: comparison of GDE3 results with simple constrained particle swarm optimization.

Solution	Simple constrained particle swarm optimization [21]	Generalized differential evolution 3
x_1	0.051583	0.0517955
x_2	0.354190	0.3592831
x_3	11.438675	11.140516
$f(\vec{x})$	0.012665	0.012665836

$$\begin{aligned} g_7(\vec{x}) &= \frac{x_2 x_3}{40} - 1 \le 0, \\ g_8(\vec{x}) &= \frac{5x_2}{x_1} - 1 \le 0, \\ g_9(\vec{x}) &= \frac{x_1}{12x_2} - 1 \le 0, \\ g_{10}(\vec{x}) &= \frac{1.5x_6 + 1.9}{x_4} - 1 \le 0, \\ g_{11}(\vec{x}) &= \frac{1.1x_7 + 1.9}{x_5} - 1 \le 0. \end{aligned} \tag{5}$$

The simple bounds of the problem are $x_1 \in [2.6, 3.6]$, $x_2 \in [0.7, 0.8]$, $x_3 \in [17, 28]$, $x_4 \in [7.3, 8.3]$, $x_5 \in [7.8, 8.3]$, $x_6 \in [2.9, 3.0]$, and $x_7 \in [5.0, 5.5]$.

The optimum design of the speed reducer is executed using GDE3 metaheuristic, and the best solution is found as $x^* = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\} = \{3.50000000047883, 0.7, 17.0, 7.3, 7.8, 3.35021466645262, 5.2866832298256\}$ which yields an objective function value of $f(\vec{x}) = 2996.34816529042$ as seen in Table 8.

The results obtained by GDE3 are presented in Table 8. GDE3 found the global optimum requiring 400 iterations per optimization run. Table 9 provides a comparison of this solution with the results of simple constrained particle swarm optimization. It is apparent from the table that GDE3 metaheuristic finds a competitive solution using only 10,000 objective function evaluations, which is considerably lesser than those of other approaches. Further, a statistical evaluation of 100 independent runs of the GDE3 metaheuristic is tabulated in Table 10 considering the best, worst, average, and the standard deviation (std. dev.) of the obtained solutions. The ratio between the optimized costs corresponding to best and worst designs is 1.0000003. Remarkably, GDE3 produced the overall best design result with a value of 2996.3481653.

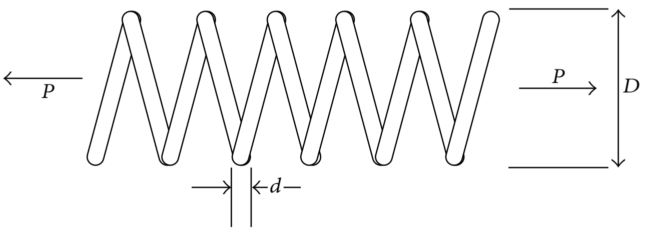


Figure 4: Schematic of the tension/compression spring design problem.

Example 4 (tension/compression spring design optimization problem). The tension/compression spring problem is designed to minimize the weight of the spring subjecting it to some constraints such as shear stress, minimum deflection, outside diameter limits, and surge frequency. The design variables are the number of active coils P, the diameter of the mean coil D, and the diameter of the wire d. Figure 4 shows the tension/compression spring design.

Assuming $x_1 = d$, $x_2 = D$, and $x_3 = P$, as the design variables, the tension/compression spring design problem can be expressed as follows:

$$\begin{aligned} \text{Minimize} \quad & f(\vec{x}) = (x_3 + 2)\, x_2 {x_1}^2, \\ \text{Subject to} \quad & g_1(\vec{x}) = 1 - \frac{{x_2}^3 x_3}{71,785{x_1}^4} \le 0, \\ & g_2(\vec{x}) = \frac{4{x_2}^2 - x_1 x_2}{12,566\,(x_2{x_1}^3 - {x_1}^4)} \\ & \qquad + \frac{1}{5,108{x_1}^2} - 1 \le 0, \\ & g_3(\vec{x}) = 1 - \frac{140.45x_1}{{x_2}^2 x_3} \le 0, \\ & g_4(\vec{x}) = \frac{x_2 + x_1}{1.5} - 1 \le 0. \end{aligned} \tag{6}$$

The simple bounds of the problem are $x_1 \in [0.05, 2.0]$, $x_2 \in [0.25, 1.3]$, and $x_3 \in [2.0, 15.0]$.

Table 13: Statistical results of the GDE3 optimization.

Best	Average	Worst	Std. dev.	Number of iterations
0.012665836	0.012666648	0.012667194	$3.97815E-07$	400

The optimum design of the tension/compression spring is carried out using GDE3 metaheuristic, and the best solution is found as $x^* = \{x_1, x_2, x_3\} = \{0.0517955276224998, 0.359283196922392, 11.1405163630287\}$ which yields an objective function value of $f(\vec{x}) = 0.0126658360085857$ as seen in Table 11.

The results obtained by GDE3 are presented in Table 11. GDE3 found the global optimum requiring 400 iterations per optimization run. Table 12 provides a comparison of this solution with the results of simple constrained particle swarm optimization. It is apparent from the table that GDE3 metaheuristic finds a competitive solution using only 10,000 objective function evaluations, which is considerably lesser than those of other approaches. Further, a statistical evaluation of 100 independent runs of the GDE3 metaheuristic is tabulated in Table 13 considering the best, worst, average, and the standard deviation (std. dev.) of the obtained solutions. The ratio between the optimized costs corresponding to worst and best designs is 1.000107.

4. Conclusion

In the present study, the GDE3 algorithm is used as a simple and efficient optimization technique for handling engineering optimization problems. The GDE3 algorithm also uses a very simple mechanism to deal with constrained functions and results generated by the algorithm indicate that such mechanism, despite its simplicity, is effective in practice. From this study, performance evaluation of the GDE3 algorithm through benchmark design optimization examples reveals the efficiency of this technique in solving practical optimization problems. Although in the present study the algorithm is utilized only for solving engineering design optimization problems, GDE3 algorithm can easily be employed for solving other types of optimization problems as well.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Mixed variable structural optimization using Firefly Algorithm," *Computers & Structures*, vol. 89, no. 23-24, pp. 2325–2336, 2011.

[2] R. Caballero, M. González, F. M. Guerrero, J. Molina, and C. Paralera, "Solving a multiobjective location routing problem with a metaheuristic based on tabu search. Application to a real case in Andalusia," *European Journal of Operational Research*, vol. 177, no. 3, pp. 1751–1763, 2007.

[3] A. Kaveh and S. Talatahari, "A novel heuristic optimization method: charged system search," *Acta Mechanica*, vol. 213, no. 3-4, pp. 267–289, 2010.

[4] O. Adekanmbi, O. Olugbara, and J. Adeyemo, "An investigation of generalized differential evolution metaheuristic for multiobjective optimal crop-mix planning decision," *The Scientific World Journal*, vol. 2014, Article ID 258749, 9 pages, 2014.

[5] F. Glover and G. A. Kochenberger, *Handbook of Metaheuristics*, Springer, 2003.

[6] M. M. Atiqullah and S. S. Rao, "Simulated annealing and parallel processing: an implementation for constrained global design optimization," *Engineering Optimization*, vol. 32, no. 5, pp. 659–685, 2000.

[7] C. A. Coello Coello, "Use of a self-adaptive penalty approach for engineering optimization problems," *Computers in Industry*, vol. 41, no. 2, pp. 113–127, 2000.

[8] J. Zhang, C. Liang, Y. Huang, J. Wu, and S. Yang, "An effective multiagent evolutionary algorithm integrating a novel roulette inversion operator for engineering optimization," *Applied Mathematics and Computation*, vol. 211, no. 2, pp. 392–416, 2009.

[9] K. E. Parsopoulos and M. N. Vrahatis, "Unified particle swarm optimization for solving constrained engineering optimization problems," in *Advances in Natural Computation*, vol. 3612 of *Lecture Notes in Computer Science*, pp. 582–591, Springer, Berlin, Germany, 2005.

[10] K. Deb, "Optimal design of a welded beam via genetic algorithms," *AIAA Journal*, vol. 29, no. 11, pp. 2013–2015, 1991.

[11] J. P. B. Leite and B. H. V. Topping, "Improved genetic operators for structural engineering optimization," *Advances in Engineering Software*, vol. 29, no. 7–9, pp. 529–562, 1998.

[12] S. Akhtar, K. Tai, and T. Ray, "A socio-behavioral simulation model for engineering design optimization," *Engineering Optimization*, vol. 34, no. 4, pp. 341–354, 2002.

[13] T. Ray and K. M. Liew, "Society and civilization: an optimization algorithm based on the simulation of social behavior," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 4, pp. 386–396, 2003.

[14] A. C. C. Lemonge and H. J. C. Barbosa, "An adaptive penalty scheme for genetic algorithms in structural optimization," *International Journal for Numerical Methods in Engineering*, vol. 59, no. 5, pp. 703–736, 2004.

[15] S. He, E. Prempain, and Q. H. Wu, "An improved particle swarm optimizer for mechanical design optimization problems," *Engineering Optimization*, vol. 36, no. 5, pp. 585–605, 2004.

[16] K. S. Lee and Z. W. Geem, "A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice," *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 36–38, pp. 3902–3933, 2005.

[17] A. R. Hedar and M. Fukushima, "Derivative-free filter simulated annealing method for constrained continuous global optimization," *Journal of Global Optimization*, vol. 35, no. 4, pp. 521–549, 2006.

[18] S.-F. Hwang and R.-S. He, "A hybrid real-parameter genetic algorithm for function optimization," *Advanced Engineering Informatics*, vol. 20, no. 1, pp. 7–21, 2006.

[19] H. S. Bernardino, H. J. C. Barbosa, and A. C. C. Lemonge, "A hybrid genetic algorithm for constrained optimization problems in mechanical engineering," in *Proceedings of the IEEE*

Congress on Evolutionary Computation (CEC '07), pp. 646–653, IEEE, 2007.

[20] M. Mahdavi, M. Fesanghary, and E. Damangir, "An improved harmony search algorithm for solving optimization problems," *Applied Mathematics and Computation*, vol. 188, no. 2, pp. 1567–1579, 2007.

[21] L. C. Cagnina, S. C. Esquivel, and C. A. Coello, "Solving engineering optimization problems with the simple constrained particle swarm optimizer," *Informatica*, vol. 32, no. 3, pp. 319–326, 2008.

[22] M. Fesanghary, M. Mahdavi, M. Minary-Jolandan, and Y. Alizadeh, "Hybridizing harmony search algorithm with sequential quadratic programming for engineering optimization problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 197, no. 33–40, pp. 3080–3091, 2008.

[23] M. Zhang, W. Luo, and X. Wang, "Differential evolution with dynamic stochastic selection for constrained optimization," *Information Sciences*, vol. 178, no. 15, pp. 3043–3074, 2008.

[24] O. Hasanтebi, S. K. Azad, and O. Hasançebi, "An efficient metaheuristic algorithm for engineering optimization: SOPT," *International Journal of Optimization in Civil Engineering*, vol. 2, no. 4, pp. 479–487, 2012.

[25] J. Golinski, "An adaptive optimization system applied to machine synthesis," *Mechanism and Machine Theory*, vol. 8, no. 4, pp. 419–436, 1973.

[26] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[27] V. L. Huang, A. K. Qin, and P. N. Suganthan, "Self-adaptive differential evolution algorithm in constrained real-parameter optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '06)*, pp. 215–222, Vancouver, Canada, July 2006.

[28] K. Suresh, D. Kundu, S. Ghosh, S. Das, and A. Abraham, "Data clustering using multi-objective differential evolution algorithms," *Fundamenta Informaticae*, vol. 97, no. 4, pp. 381–403, 2009.

[29] S. Kukkonen and J. Lampinen, "Generalized differential evolution for general non-linear optimization," in *COMPSTAT 2008*, pp. 459–471, Springer, 2008.

[30] J. Rönkkönen, S. Kukkonen, and J. Lampinen, "A comparison of differential evolution and generalized generation gap model," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 9, no. 5, pp. 549–555, 2005.

[31] S. Kukkonen and J. Lampinen, "GDE3: the third evolution step of generalized differential evolution," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '05)*, pp. 443–450, September 2005.

[32] G. G. Dimopoulos, "Mixed-variable engineering optimization based on evolutionary and social metaphors," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, no. 4, pp. 803–817, 2007.

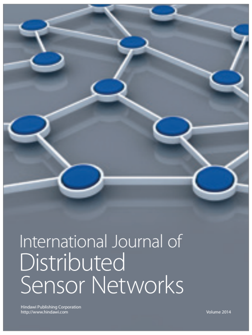
The Scientific World Journal

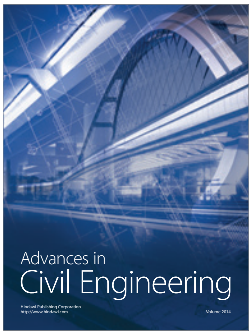
International Journal of
Rotating Machinery

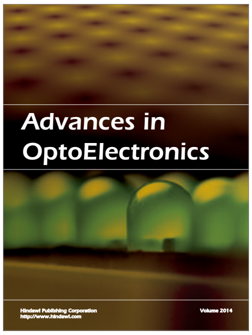
Journal of
Engineering

Journal of
Sensors

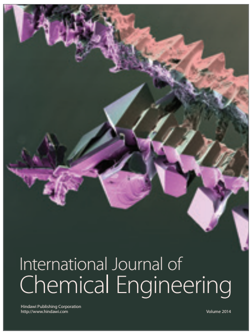
Advances in
Mechanical
Engineering


International Journal of
Distributed
Sensor Networks


Advances in
Civil Engineering

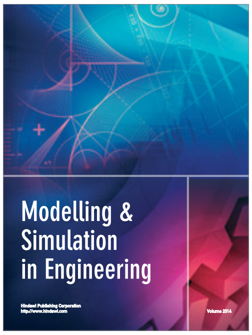

Advances in
OptoElectronics

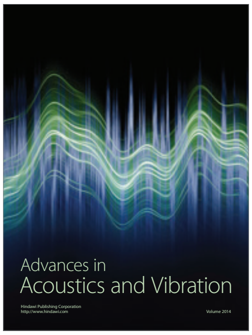
Journal of
Robotics


International Journal of
Chemical Engineering

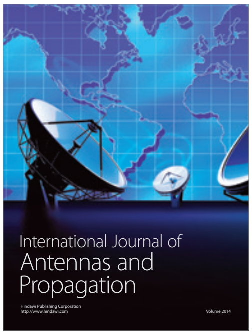
VLSI Design

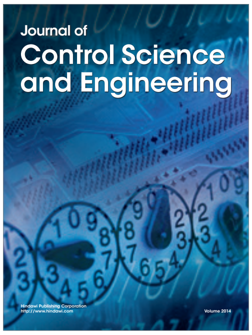
International Journal of
Navigation and
Observation

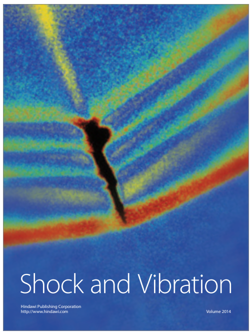

Modelling &
Simulation
in Engineering


Advances in
Acoustics and Vibration

Active and Passive
Electronic Components


International Journal of
Antennas and
Propagation


Journal of
Control Science
and Engineering


Shock and Vibration

Journal of
Electrical and Computer
Engineering