Engineering Optimization

ISSN: 0305-215X (Print) 1029-0273 (Online) Journal homepage: http://www.tandfonline.com/loi/geno20

A new optimization algorithm for solving complex constrained design optimization problems

R. Venkata Rao & G.G. Waghmare

To cite this article: R. Venkata Rao & G.G. Waghmare (2016): A new optimization algorithm for solving complex constrained design optimization problems, Engineering Optimization, DOI: 10.1080/0305215X.2016.1164855

To link to this article: http://dx.doi.org/10.1080/0305215X.2016.1164855

Published online: 13 Apr 2016.

ENGINEERING OPTIMIZATION, 2016
http://dx.doi.org/10.1080/0305215X.2016.1164855

A new optimization algorithm for solving complex constrained design optimization problems

R. Venkata Rao and G.G. Waghmare

Department of Mechanical Engineering, S.V. National Institute of Technology, Surat, India

ABSTRACT
This article presents the performance of a very recently proposed Jaya algorithm on a class of constrained design optimization problems. The distinct feature of this algorithm is that it does not have any algorithm-specific control parameters and hence the burden of tuning the control parameters is minimized. The performance of the proposed Jaya algorithm is tested on 21 benchmark problems related to constrained design optimization. In addition to the 21 benchmark problems, the performance of the algorithm is investigated on four constrained mechanical design problems, *i.e.* robot gripper, multiple disc clutch brake, hydrostatic thrust bearing and rolling element bearing. The computational results reveal that the Jaya algorithm is superior to or competitive with other optimization algorithms for the problems considered.

ARTICLE HISTORY
Received 30 November 2015
Accepted 25 February 2016

KEYWORDS
Constrained design optimization; benchmark problems; Jaya algorithm

1. Introduction

Design optimization problems are important from both an industrial and a scientific perspective. In practical design problems and situations, there are more design variables and their influence on the objective function is significant. Moreover, the designer requires a global optimum, whereas the objective function may become trapped in local optima. Classical methods are not efficient in solving such problems since they compute only local optima. Therefore, an intelligent method is required for solving constrained design problems effectively.

The field of optimization has grown rapidly during the past few decades. Many new theoretical, algorithmic and computational contributions of optimization have been proposed to solve various problems in engineering and management. Recent developments of optimization methods can be mainly divided into deterministic and heuristic approaches. Deterministic approaches take advantage of the analytical properties of the problem to generate a sequence of points that converge to a global optimal solution. Deterministic approaches (*e.g.* linear programming, nonlinear programming and mixed-integer nonlinear programming) can provide general tools for solving optimization problems to obtain a global or an approximately global optimum (Lin, Tsai, and Yu 2012). For solving non-convex or large-scale optimization problems, it may not be easy to derive a globally optimal solution within a reasonable time using deterministic methods, owing to the high complexity of the problem. Heuristic approaches have been found to be more flexible and efficient than deterministic approaches and reduce the computational time taken to solve an optimization problem, but the obtained solution is not guaranteed to be a feasible or globally optimal solution.

Constrained design optimization problems have been addressed by various researchers using different techniques such as the genetic algorithm (GA) (Goldberg 1989), particle swarm optimization

CONTACT G.G. Waghmare waghmaregaju@yahoo.com

(PSO) (Kennedy and Eberhart 1995; Clerc 2006; Liu, Cai, and Wang 2010), artificial bee colony (ABC) (Karaboga 2005; Karaboga and Basturk 2007), ant colony optimization (ACO) (Dorigo and Stutzle 2004; Blum 2005), harmony search (HS) (Lee and Geem 2004) and the grenade-explosion method (GEM) (Ahrari and Atai 2010). These algorithms have been effectively used for solving many optimization problems. Lee and Geem (2004) attempted unconstrained, constrained and structural optimization problems using the HS algorithm. Parsopoulos and Vrahatis (2005) solved four well-known constrained design optimization problems (*i.e.* tension/compression spring, welded beam, gear train and pressure vessel) using a unified particle swarm optimization (UPSO) method. Mezura-Montes and Coello (2005) solved engineering design problems without using a penalty function by introducing an evolutionary algorithm to maintain infeasible solutions close to the feasible region. Becerra and Coello (2006) solved constrained optimization problems by proposing a cultural algorithm using differential evolution (CDE). Huang, Wang, and He (2007) solved constrained optimization problems using a co-evolutionary differential evolution (CoDE) algorithm. He and Wang (2007) effectively tackled constrained optimization problems employing a co-evolutionary model in the particle swarm optimization technique (CPSO). Liao (2010) tested the performance of two hybrid differential evolution (DE) algorithms on 14 engineering design problems. Akay and Karaboga (2010) tested engineering design problems using the ABC algorithm. Rao, Savsani, and Vakharia (2011) proposed and tested the performance of the teaching–learning-based optimization (TLBO) algorithm on design optimization problems. Zhang *et al.* (2013) combined a tissue membrane system and DE in their differential evolution algorithm and tissue P systems (DETPS) algorithm and solved constrained design optimization problems. Pavone, Narzisi, and Nicosia (2012) presented an immunological algorithm to solve global numerical optimization problems for high-dimensional instances. The authors designed two versions of the immunological algorithms: the first based on binary-code representation and the second based on real values. Brajevic and Tuba (2013) tackled constrained design optimization problems using an upgraded ABC algorithm. Rios and Sahinidis (2013) addressed the solution of bound-constrained optimization problems using a derivative-free algorithm which required only the availability of objective function values but no derivative information. The authors provided a review of derivative-free algorithms, followed by a systematic comparison of 22 related implementations using a test set of 502 problems. Rao and Waghmare (2014) tested the performance of an elitist teaching–learning-based optimization algorithm (Elitist TLBO) on constrained design problems. The TLBO algorithm has gained wide acceptance among optimization researchers.

In view of the success of the TLBO algorithm, Rao (2016) very recently proposed another algorithm-specific parameter-free algorithm named the Jaya algorithm. The Jaya algorithm requires only the common control parameters and not the algorithm-specific parameters. Common control parameters such as population size, number of generations and elite size are common to running any of the optimization algorithms, whereas algorithm-specific parameters are specific, *i.e.* different algorithms have different specific parameters to control. The other evolutionary algorithms require control of common control parameters as well as their own algorithm-specific parameters. For example, the GA requires the tuning of specific parameters such as crossover probability, mutation probability and selection operator, in addition to tuning the common control parameters. Similarly, the PSO algorithm requires tuning of specific parameters including inertia weight, social factors and cognitive factors, in addition to the tuning of common control parameters. Thus, the burden of tuning of control parameters is reduced in the Jaya algorithm as it requires tuning of only the common control parameters. Unlike the two phases (*i.e.* teacher phase and learner phase) of the TLBO algorithm, the Jaya algorithm has only one phase and is comparatively simple to apply. The working of the Jaya algorithm is different from that of the TLBO algorithm. In the previous work by Rao (2016), the Jaya algorithm was tested on unconstrained and constrained benchmark problems. However, to check the effectiveness of any new proposed algorithm, its performance on a number of complex benchmark problems must be assessed. In this article, the performance of the Jaya algorithm is investigated by extending the evaluation through experimentation with 21 constrained benchmark design optimization problems and four mechanical design optimization problems. To validate the proposed algorithm, its results are compared with the results of different algorithms for different benchmark functions. The following section presents a brief description of the Jaya algorithm (Rao 2016).

2. Jaya algorithm

Let $f(x)$ be the objective function to be minimized (or maximized). At any iteration i, assume that there are m design variables and n candidate solutions (*i.e.* population size, $k = 1,2, \ldots ,n$). Let the best candidate, 'best', obtain the best value of $f(x)$ (*i.e.* $f(x)_{\text{best}}$) in the entire candidate solutions and the worst candidate, 'worst', obtain the worst value of $f(x)$ (*i.e.* $f(x)_{\text{worst}}$) in the entire candidate solutions. If $X_{j,k,i}$ is the value of the jth variable for the kth candidate during the ith iteration, then this value is modified as per Equation (1):

$$X'_{j,k,i} = X_{j,k,i} + r_{1,j,i}(X_{j,\text{best},i} - |X_{j,k,i}|) - r_{2,j,i}(X_{j,\text{worst},i} - |X_{j,k,i}|) \tag{1}$$

where $X_{j,\text{best},i}$ is the value of the variable j for the best candidate and $X_{j,\text{worst},i}$ is the value of the variable j for the worst candidate. $X'_{j,k,i}$ is the updated value of $X_{j,k,i}$, and $r_{1,j,i}$ and $r_{2,j,i}$ are the two random numbers for the jth variable during the ith iteration in the range [0, 1]. The term $r_{1,j,i}$ $((X_{j,\text{best},i} - X_{j,k,i})$ indicates the tendency of the solution to move closer to the best solution, and the term $-r_{2,j,i}$ $(X_{j,\text{worst},i} - X_{j,k,i}|)$ indicates the tendency of the solution to avoid the worst solution. $X'_{j,k,i}$ is accepted if it gives a better function value. All the accepted function values at the end of the iteration are maintained and these values become the input to the next iteration. In the proposed algorithm, the solution obtained for a given problem is moving towards the best solution and avoiding the worst solution. The random numbers r_1 and r_2 ensure good exploration of the search space. The absolute value of the candidate solution ($|X_{j,k,i}|$) considered in Equation (1) further enhances the exploration ability of the algorithm. Figure 1 shows the flowchart of the Jaya algorithm.

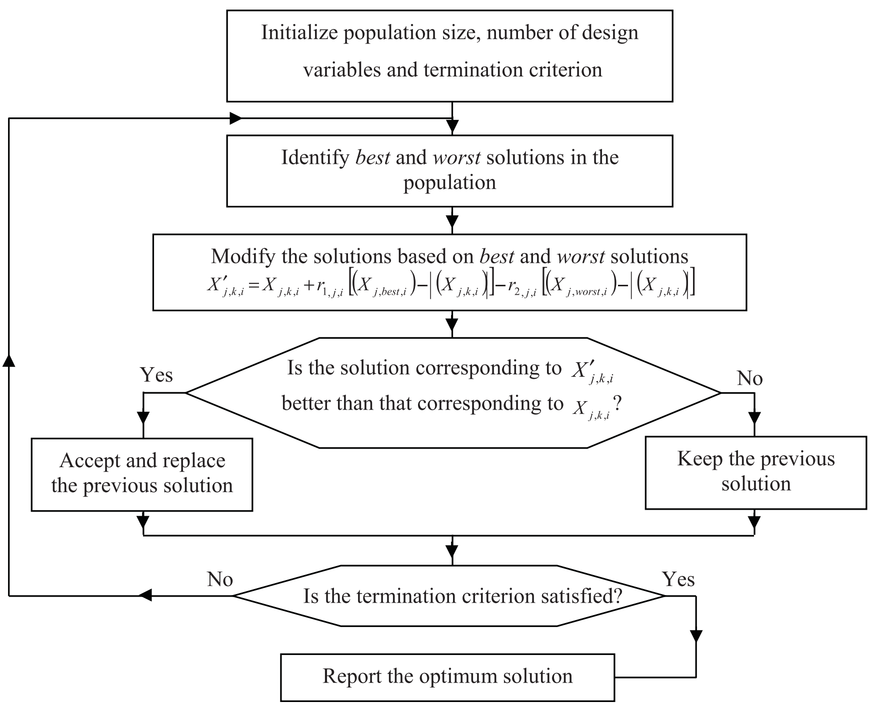

Figure 1. Flowchart of the Jaya algorithm (Rao 2016).

The next section deals with the experimentation of the Jaya algorithm on various constrained benchmark functions and design optimization problems.

3. Experiments on constrained benchmark optimization problems

In this section, 21 constrained design benchmark optimization problems and four mechanical design problems from the literature are used to test the performance of the Jaya algorithm. Various researchers had attempted these problems using different optimization algorithms. Now, the Jaya algorithm is applied to the same problems and comparisons are made. One of the major issues of constrained optimization is how to deal with infeasible individuals throughout the search space, using procedures commonly known as constraint handling methods. Constraint handling methods are categorized based on different handling techniques, such as methods based on penalty functions, methods based on searching for feasible solutions, methods based on preserving the feasibility of solutions and hybrid methods. Deb's heuristic constrained handling method (Deb 2000) is used to handle the constraints in the present experiments. This method transforms a constrained optimization problem into an unconstrained problem by adding a penalty factor to the fitness value of each infeasible individual so that it is penalized for violating one or more of the constraints. The statistical results on constrained design benchmark problems using the Jaya algorithm and its comparison with other optimization algorithms are explained in detail in the following section.

4. Experimental results and discussion

Details of the 21 constrained design benchmark optimization problems and four mechanical design problems are presented in Appendix 1. Problem 1 is a minimization problem and the objective function involves 13 variables and nine linear inequality constraints. The Jaya algorithm is applied to solve this problem and its performance is compared with eight other optimization algorithms. Just like any other algorithm, the Jaya algorithm requires proper tuning of common control parameters such as population size and number of generations to execute the algorithm effectively. After running a few trials, a population size of 50 and number of function evaluations (NFE) of 1500 are considered. The results of Elitist TLBO, DETPS, TLBO, multimembered evolution strategy (M-ES), particle evolutionary swarm optimization (PESO), CDE, CoDE and ABC are taken from Rao and Waghmare (2014), Zhang *et al.* (2013), Rao, Savsani, and Vakharia (2011), Mezura-Montes and Coello (2005), Zavala, Aguirre, and Diharce (2005), Becerra and Coello (2006), Huang, Wang, and He (2007) and Karaboga and Basturk (2007), respectively.

Problem 2 is a maximization problem involving 10 variables and a nonlinear constraint. After running a few trials, a population size of 50 and function evaluations of 25,000 are considered. Problem 3 is a minimization problem which involves seven variables and four nonlinear inequality constraints. After running a few trials, a population size of 10 and NFE of 30,000 are considered.

Problem 4 is a linear minimization problem which involves eight variables and three nonlinear inequality and three linear inequality constraints. After running a few trials, a population size of 10 and NFE of 99,000 are considered. The convergence plot of the proposed algorithm for problem 4 is presented in Figure 2.

Problem 5 is a maximization problem involving three design variables and 729 nonlinear inequality constraints. After running a few trials, a population size of 50 and NFE of 5000 are considered. The comparison of statistical results of the considered nine algorithms for test problems 1–5 is presented in Table 1, where 'Best', 'Mean' and 'Worst' represent the best solution, mean best solution and worst solution, respectively, over 30 independent runs.

Problem 6 is minimization of cost for a welded beam design having four continuous design variables as shown in Figure 3. A population size of 10 and NFE of 10,000 are used. Problem 7 is minimization of total cost for a pressure vessel design involving three linear constraints and one

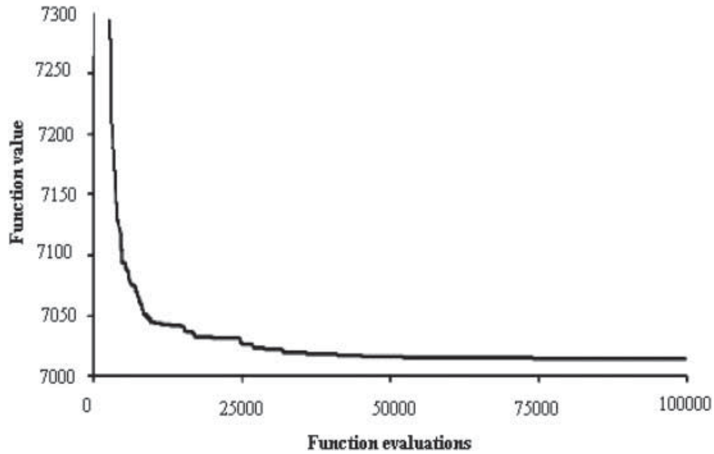

Figure 2. Convergence plot of the proposed algorithm for problem 4.

nonlinear inequality constraint, and two discrete and two continuous design variables, as shown in Figure 4. A population size of 10 and NFE of 10,000 are considered.

Problem 8 is minimization of weight for tension/compression spring design involving one linear constraint and three nonlinear inequality constraints with three continuous design variables. The tension/compression spring is shown in Figure 5. A population size of 10 and NFE of 10,000 are considered. Problem 9 is minimization of weight for a speed reducer design involving one discrete and six continuous design variables, and four linear and seven nonlinear inequality constraints, as shown in Figure 6. After running a few trials, a population size of 10 and NFE of 10,000 are considered. Table 2 presents the statistical results of 11 algorithms for test problems 6–9 over 30 independent runs.

For problems 10, 11 and 12, population sizes of 10, 100, 50 and NFE of 4000, 79,000, 5000, respectively, are considered after running a few trials. Problems 13, 14, 15 and 16 are process synthesis design problems, for which population sizes of 10, 10, 10, 10 and NFE of 900, 500, 1000, 7000, respectively, are considered. For problems 17, 18 and 19, population sizes of 10, 10, 10 and NFE of 300, 2900, 100, respectively, are considered.

Problem 20 is minimization of total cost for a pressure vessel design. A population size of 50 and NFE of 5000 are considered. Problem 21 is maximization of material removal rate for a ceramic grinding process. A population size of 10 and NFE of 1000 are considered.

For each of the above problems, 30 independent runs are carried out and the NFE is recorded in Table 3. Table 4 presents the statistical results of the mean best solution and standard deviation for six optimization algorithms.

In addition to the 21 benchmark problems, the performance of the Jaya algorithm is investigated on four constrained mechanical design problems, namely a robot gripper, multiple disc clutch brake, hydrostatic thrust bearing and rolling element bearing, taken from Rao, Savsani, and Vakharia (2011). Table 5 presents the comparison of results for these mechanical design problems obtained using the Jaya, TLBO and ABC algorithms. The robot gripper design is related to minimization of the objective function of the difference between the maximum and minimum force applied by the gripper for the range of gripper end displacements problem involving seven continuous design variables and six different constraints. After running a few trials, a population size of 50 and NFE of 25,000 are considered. The multiple disc clutch brake problem involves minimization of the mass of multiple disc clutch brakes as an objective function using the inner radius, outer radius, thickness of discs, actuating force and number of friction surfaces as five discrete variables. After running a few trials, a population size of 20 and NFE of 600 are considered. The minimization of the power loss is considered as

Table 1. Comparison of statistical results of nine algorithms for test problems 1–5.

Problem		Jaya algorithm	Elitist TLBO (Rao and Waghmare 2014)	DETPS (Zhang *et al.* 2013)	TLBO (Rao, Savsani, and Vakharia 2011)	M-ES (Mezura-Montes and Coello 2005)	PESO (Zavala, Aguirre, and Diharce 2005)	CDE (Becerra and Coello 2006)	CoDE (Huang, Wang, and He 2007)	ABC (Karaboga and Basturk 2007)
1	Best	**−15.0**	**−15.0**	**−15.0**	**−15.0**	**−15.0**	**−15.0**	**−15.0**	**−15.0**	**−15.0**
	Mean	**−15.0**	**−15.0**	**−15.0**	**−15.0**	**−15.0**	**−15.0**	**−15.0**	**−15.0**	**−15.0**
	Worst	**−15.0**	**−15.0**	**−15.0**	**−15.0**	**−15.0**	**−15.0**	**−15.0**	**−15.0**	**−15.0**
	SD	**0.000**	1.9e-6	–	–	–	–	–	–	–
	NFE	75,000	**14,009**	20,875	25,000	240,000	350,000	100,100	2,48,000	240,000
2	Best	**1.000**	**1.000**	**1.001**	**1.000**	**1.000**	**1.005**	0.995	–	**1.000**
	Mean	**1.000**	**1.000**	0.992	**1.000**	**1.000**	**1.005**	0.789	–	**1.000**
	Worst	**1.000**	**1.000**	0.995	**1.000**	**1.000**	**1.005**	0.640	–	**1.000**
	SD	**0.000**	**0.000**	–	–	–	–	–	–	–
	NFE	**25,000**	69,996	90,790	100,000	240,000	350,000	100,100	–	240.000
3	Best	**680.630**	**680.630**	**680.630**	**680.630**	680.632	**680.630**	**680.630**	680.771	680.634
	Mean	680.639	680.631	**680.630**	680.633	680.643	**680.630**	**680.630**	681.503	680.640
	Worst	680.651	680.633	**680.630**	680.638	680.719	**680.630**	**680.630**	685.144	680.653
	SD	**2.9e-2**	3.4e-3	–	–	–	–	–	–	–
	NFE	**30,000**	30,019	32,586	100,000	240,000	350,000	100,100	240,000	100,000
4	Best	**7049.248**	**7049.248**	7049.257	**7049.248**	7051.903	7049.459	**7049.248**	–	7053.904
	Mean	7056.632	7050.261	7050.834	7083.673	7253.047	7099.101	**7049.248**	–	7224.407
	Worst	7087.620	7055.481	7063.406	7224.497	7099.101	7251.396	**7049.248**	–	7604.132
	SD	**2.6e-2**	2.8e-2	–	–	–	–	–	–	–
	NFE	**99,000**	99,987	100,000	100,000	240,000	350,000	100,100	–	240,000
5	Best	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
	Mean	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
	Worst	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
	SD	**0.000**	**0.000**	–	–	–	–	–	–	–
	NFE	**5000**	5011	6540	50,000	240,000	350,000	100,100	240,000	100,000

Note: Elitist TLBO = elitist teaching-learning-based optimization; DETPS = differential evolution algorithm and tissue P systems; TLBO = teaching-learning-based optimization; M-ES = multimembered evolution strategy; PESO = particle evolutionary swarm optimization; CDE = cultural differential evolution; CoDE = co-evolutionary differential evolution; ABC = artificial bee colony; NFE = number of function evaluations; – = result is not available.

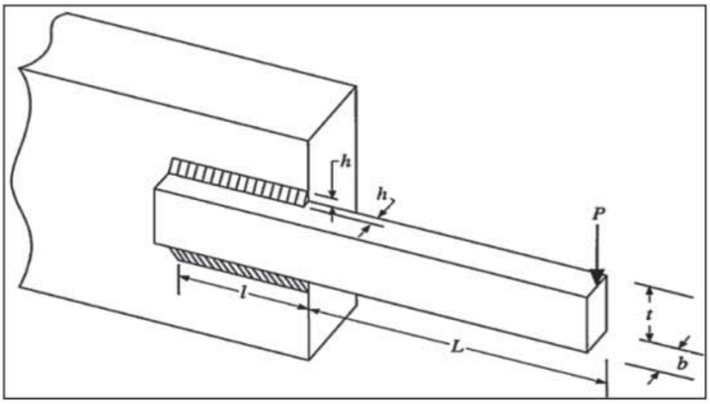

Figure 3. The welded beam design problem (Rao, Savsani, and Vakharia 2011).

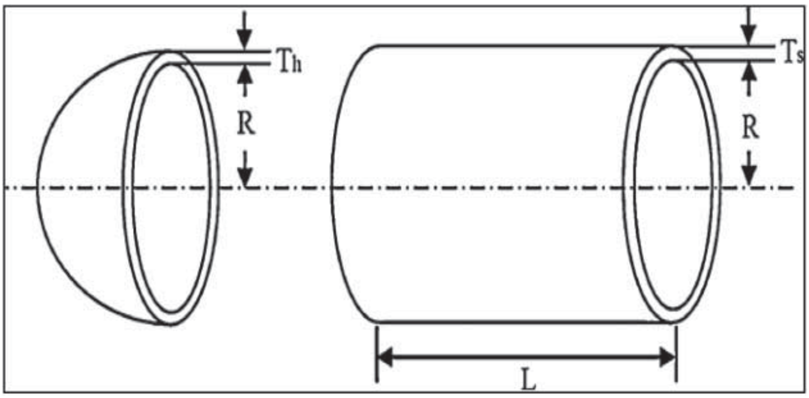

Figure 4. Centre and end section of the pressure vessel design problem (Rao, Savsani, and Vakharia 2011).

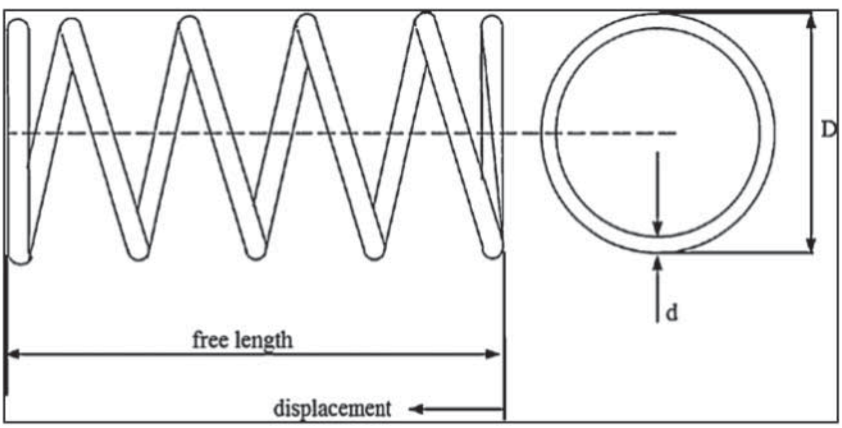

Figure 5. The tension/compression spring design problem (Rao, Savsani, and Vakharia 2011).

an objective function for a hydrostatic thrust bearing containing four design variables and seven different constraints. A population size of 50 and NFE of 25,000 are considered. Maximization of the dynamic load-carrying capacity of a rolling element bearing is considered as an objective function

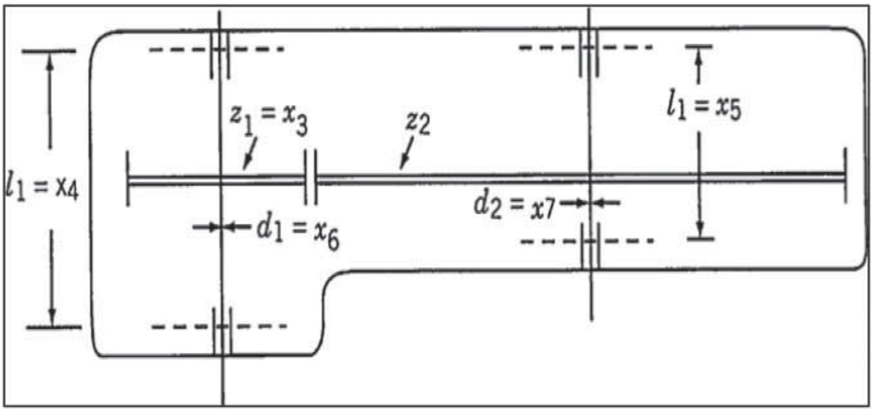

Figure 6. The speed reducer problem (Rao, Savsani, and Vakharia 2011).

involving five design variables, namely, ball diameter, pitch diameter, inner and outer raceway curvature coefficients, and number of balls. After running a few trials, a population size of 50 and NFE of 10,000 are considered. A system with an Intel core i3 2.53 GHz processor and 1.85 GB RAM is used for implementing the MATLAB code for the Jaya algorithm on the considered problems.

The statistical results of nine algorithms for test problems 1–5 are shown in Table 1. For problem 1, the Jaya algorithm finds the same global optimum solution as that given by the Elitist TLBO, DETPS, TLBO, M-ES, PESO, CDE, CoDE and ABC algorithms, but the Jaya algorithm uses 31.25%, 21.42%, 74.92%, 30.24% and 31.25% function evaluations to obtain competitive results compared to M-ES, PESO, CDE, CoDE and ABC, respectively, while the Elitist TLBO requires approximately 94%, 94%, 86%, 96%, 94%, 44% and 33% function evaluations compared to ABC, CoDE, CDE, PESO, M-ES, TLBO and DETPS, respectively. The Jaya algorithm is superior to the Elitist TLBO algorithm in terms of standard deviation and robustness for problem 1.

For problem 2, the Jaya algorithm finds a better quality of solution than DETPS and CDE. The results of the Jaya algorithm are same as those of the Elitist TLBO, TLBO, M-ES and ABC algorithms, but it uses 35.71%, 27.53%, 25%, 10.41%, 7.41%, 24.97% and 10.41% function evaluations to obtain competitive results compared to Elitist TLBO, DETPS, TLBO, M-ES, PESO, CDE, and ABC, respectively. The Jaya algorithm also computes the function value in less time than the other optimization algorithms considered.

For problem 3, the results of the Jaya algorithm are same as those of the Elitist TLBO, DETPS, TLBO, PESO and CDE algorithms in terms of the best solution, but the Jaya algorithm uses 99.23%, 92.06%, 30%, 12.5%, 8.57%, 29.97%, 12.5% and 30% function evaluations to obtain competitive results compared to Elitist TLBO, DETPS, TLBO, M-ES, PESO, CDE, CoDE and ABC, respectively. For problem 3, the solutions given by the Jaya algorithm, Elitist TLBO, M-ES, CoDE and ABC algorithms are inferior to those given by DETPS, PESO and CDE in terms of mean and worst solutions. However, to obtain the global optimum solution, DETPS, PESO and CDE needed more function evaluations than the Jaya algorithm.

For problem 4, the results of the Jaya algorithm are same as those of the Elitist TLBO, TLBO and CDE algorithms in terms of the best solution, but the Jaya algorithm uses 99.01%, 99% and 98.90% function evaluations to obtain competitive results compared to Elitist TLBO, TLBO and CDE, respectively. However, the solution given by CDE is better than those given by the Jaya algorithm, Elitist TLBO, DETPS, TLBO, M-ES, PESO, CoDE and ABC algorithms in terms of mean and worst solutions. However, CDE requires more function evaluations than the Jaya algorithm to obtain the global optimum solution. The Jaya algorithm uses 99.01%, 99%, 99%, 41.25%, 28.28%, 98.90% and 41.25% function evaluations to obtain competitive results compared to Elitist TLBO, DETPS, TLBO, M-ES, PESO, CDE and ABC, respectively.

Table 2. Comparison of statistical results of 11 algorithms for test problems 6–9.

Problem		Jaya algorithm	Elitist TLBO (Rao and Waghmare 2014)	DETPS (Zhang *et al.* 2013)	$(\mu+\lambda)$-ES (Mezura-Montes and Coello 2005)	UPSO (Parsopoulos and Vrahatis 2005)	CPSO (He and Wang, 2007)	CoDE (Huang, Wang, and He 2007)	PSO-DE (Liu, Cai, and Wang 2010)	ABC (Akay and Karaboga 2010)	TLBO (Rao, Savsani, and Vakharia 2011)	MBA (Sadollah *et al.* 2013)
Welded beam	Best	**1.724852**	**1.724852**	**1.724852**	**1.724852**	1.92199	1.728024	1.733462	1.724853	**1.724852**	**1.724852**	1.724853
	Mean	**1.724852**	**1.724852**	**1.724852**	1.777692	2.83721	1.748831	1.768158	1.724858	1.741913	1.728447	1.724853
	Worst	**1.724853**	**1.724853**	**1.724853**	2.074562	4.88360	1.782143	1.824105	1.724881	–	–	**1.724853**
	SD	3.3e-2	3.3e-2	2.1e-7	8.8e-2	6.8e-1	1.3e-2	2.2e-2	4.1e-6	3.1e-2	–	**6.9e-19**
	NFE	10,000	**9991**	10,000	30,000	100,000	200,000	240,000	33,000	30,000	10,000	47,340
Pressure vessel	Best	**5885.3336**	**5885.3336**	**5885.3336**	6059.7016	6544.27	6061.0777	6059.7340	6059.7143	6059.7147	6059.7143	5889.3216
	Mean	**5885.3338**	5887.3338	5887.3161	6379.9380	9032.55	6147.1332	6085.2303	6059.7143	6245.3081	6059.7143	6200.64765
	Worst	**5885.805**	5956.6921	5942.3234	6820.3975	11638.20	6363.8041	6371.0455	6059.7143	–	–	6392.5062
	SD	1.0e+1	1.1e+1	1.0e+1	2.1e+2	9.9e+2	8.6e+1	4.3e+1	**1.0e-10**	2.1e+2	–	160.34
	NFE	10,000	**4992**	10,000	30,000	100,000	200,000	240,000	42,100	30,000	10,000	70,650
Tension compression spring	Best	**0.012665**	**0.012665**	**0.012665**	0.012689	0.013120	0.012675	0.012670	**0.012665**	**0.012665**	**0.012665**	**0.12665**
	Mean	**0.012666**	0.012678	0.012680	0.013165	0.022948	0.012730	0.012703	0.012665	0.012709	0.012666	0.012713
	Worst	0.012679	0.012758	0.012769	0.014078	0.050365	0.012924	0.012790	**0.012665**	–	–	0.012900
	SD	4.9e-4	4.9e-4	2.7e-5	3.9e-4	7.2e-3	5.2e-5	2.7e-5	**1.2e-8**	1.3e-2	–	6.3e-5
	NFE	10,000	**7022**	10,000	30,000	100,000	200,000	240,000	24,950	30,000	10,000	7650
Speed reducer	Best	**2996.348**	2996.348	2996.348	**2996.348**	–	–	–	2996.348	2997.058	2996.348	**2994.74421**
	Mean	**2996.348**	**2996.348**	**2996.348**	**2996.348**	–	–	–	**2996.348**	2997.058	**2996.348**	2996.769019
	Worst	**2996.348**	**2996.348**	**2996.348**	**2996.348**	–	–	–	**2996.348**	–	–	2999.6524
	SD	**0.0**	4.5e-5	5.2e-5	**0.0**	–	–	–	6.4e-6	**0.0**	–	1.56
	NFE	10,000	9.988	10,000	30,000	–	–	–	54,350	30,000	10,000	**6300**

Note: Elitist TLBO = elitist teaching-learning-based optimization; DETPS = differential evolution algorithm and tissue P systems; $(\mu+\lambda)$-ES = $(\mu+\lambda)$-evolutionary strategy; UPSO = unified particle swarm optimization; CPSO = co-evolutionary particle swarm optimization; CoDE = co-evolutionary differential evolution; PSO-DE = hybridizing particle swarm optimization with differential evolution; ABC = artificial bee colony; TLBO = teaching-learning-based optimization; MBA = mine blast algorithm; SD = standard deviation; NFE = number of function evaluations; - = result is not available.

Table 3. Number of function evaluations of five algorithms for test problems 10–21.

Problem	maxNFE	Jaya	Elitist TLBO (Rao and Waghmare 2014)	DETPS (Zhang *et al.* 2013)	MDE (Liao 2010)	MA-MDE (Liao 2010)	MDE-IHS (Liao 2010)
10	15,000	**4,000**	4,033	14,360	7,777	4,436	5,359
11	100,000	**79,000**	79,994	100,000	96,718	93,524	83,442
12	15,000	**5,000**	5,029	7,254	7,688	13,023	14,518
13	5,000	**900**	1,019	1,720	1,075	1,430	3,297
14	5,000	**500**	599	1,726	827	653	1,409
15	50,000	**1,000**	4,112	5,309	30,986	25,766	22,146
16	50,000	**7,000**	7,987	8,907	37,739	20,116	27,1163
17	5,495	**300**	312	3,242	1,240	1,955	493
18	50,000	**2,900**	2,989	4,597	21,539	35,180	9,733
19	1,000	**100**	105	144	333	449	176
20	50,000	**5,000**	5,012	5,054	46,868	39,902	38,237
21	10,000	**1,000**	1,499	9,181	1,679	2,827	4,266

Note: maxNFE = maximum number of function evaluations as the termination criterion; Elitist TLBO = elitist teaching–learning-based optimization; MDE = modified differential evolution; MA-MDE = modified differential evolution–local search hybrid; MDE-IHS = modified differential evolution–improved harmony search hybrid.

Table 4. Comparison of statistical results of mean best solution (MBS) and standard deviation (SD) for test problems 10–21.

Problem		Jaya	Elitist TLBO (Rao and Waghmare 2014)	DETPS (Zhang *et al.* 2013)	MDE (Liao 2010)	MA-MDE (Liao 2010)	MDE-IHS (Liao 2010)
10	MBS	**87.500012**	87.50002	87.500012	89.879034	88.230145	87.497550
	SD	0.000299	0.000299	**0.000002**	2.768746	1.899683	0.002118
11	MBS	**7.6671**	7.815678	7.931112	7.918619	7.883841	7.848896
	SD	**0.0000**	0.056831	**0.000000**	0.047891	0.098982	0.121909
12	MBS	**4.579592**	**4.579592**	4.579593	4.661414	4.579595	4.579599
	SD	0.000017	0.000029	**0.000000**	0.311365	0.000003	0.000005
13	MBS	**2.000000**	**2.000000**	**2.000000**	2.009348	**2.000000**	2.000001
	SD	**0.000000**	**0.000000**	**0.000000**	0.043579	**0.000000**	0.000000
14	MBS	2.124701	2.124781	2.182346	2.167894	**2.124574**	2.124604
	SD	0.000098	0.000098	0.149789	0.132196	**0.000071**	0.000076
15	MBS	**1.076599**	**1.076599**	1.076700	1.124453	1.099805	1.094994
	SD	**0.000389**	0.000409	0.000391	0.075163	0.055618	0.952898
16	MBS	3.567464	**3.557493**	3.557574	3.599903	3.564912	3.561157
	SD	0.000678	0.000587	**0.000275**	0.059012	0.029017	0.008381
17	MBS	**−32,217.430718**	**−32,217.430718**	−32,217.407346	−32,217.427262	−32,217.427106	−32,217.427780
	SD	**0.000000**	**0.000000**	0.005418	0.002836	0.003690	**0.000000**
18	MBS	**−0.808844**	**−0.808844**	**−0.808844**	−0.807608	−0.807907	**−0.808844**
	SD	0.000378	0.000397	**0.000000**	0.005615	0.003077	**0.000000**
19	MBS	**−0.974565**	**−0.974565**	**−0.974565**	−0.974505	−0.974335	**−0.974565**
	SD	0.0000247	0.0000267	**0.000000**	0.000330	0.000977	**0.000000**
20	MBS	5850.77012	5850.770128	**5850.723228**	6070.604982	6040.005940	6082.551078
	SD	89.573821	141.163893	**0.032991**	109.163780	168.603518	185.056741
21	MBS	−75.134101	−75.134130	**−75.136711**	−75.134137	−75.134130	−75.134137
	SD	0.000037	0.000046	**0.0000010**	0.000023	0.000024	0.000025

Note: Elitist TLBO = elitist teaching–learning-based optimization; DETPS = differential evolution algorithm and tissue P systems; MDE = modified differential evolution; MA-MDE = modified differential evolution–local search hybrid; MDE-IHS = modified differential evolution–improved harmony search hybrid.

For problem 5, the Jaya algorithm finds the same global optimum solution as the Elitist TLBO, DETPS, TLBO, M-ES, PESO, CDE, CoDE and ABC algorithms, but Jaya uses 99.78%, 76.45%, 10%, 2.08%, 1.42%, 4.99%, 2.08% and 5% function evaluations to obtain competitive results compared to Elitist TLBO, DETPS, TLBO, M-ES, PESO, CDE, CoDE and ABC, respectively.

The statistical results of 11 algorithms for test problems 6–9 are compared in Table 2. For the welded beam (problem 6), the results of the Jaya algorithm, Elitist TLBO and DETPS are better than

Table 5. Comparison of results for mechanical design problems obtained using the Jaya, teaching–learning-based optimization (TLBO) and artificial bee colony (ABC) algorithms for the four mechanical design problems.

Problem		Jaya	TLBO (Rao, Savsani, and Vakharia 2011)	ABC (Rao, Savsani, and Vakharia 2011)
Robot gripper	Best	**4.247644**	**4.247644**	**4.247644**
	Mean	**4.93765920**	4.93770095	5.086611
	Worst	8.07837533	8.141973	**6.784631**
	SR	**0.59**	0.56	0.07
Multiple disc clutch brake	Best	**0.313657**	**0.313657**	**0.313657**
	Mean	**0.324425**	0.3271662	0.324751
	Worst	0.3867384	0.392071	**0.352864**
	SR	**0.69**	0.67	0.54
Hydrostatic thrust bearing	Best	**1625.44271**	1625.443	1625.44276
	Mean	**1796.89367**	1797.70798	1861.554
	Worst	2104.3776	**2096.80127**	2144.836
	SR	**0.21**	0.19	0.05
Rolling element bearing	Best	**81,859.7419**	81,859.74	81,859.7416
	Mean	81,439.712	81,438.987	**81,496**
	Worst	**80,808.46758**	80,807.8551	78,897.81
	SR	**0.70**	0.66	0.69

Note: SR = success rate.

those of the other algorithms. The results of the mine blast algorithm (MBA) are better those of the other nine optimization algorithms in terms of standard deviation. However, the Jaya algorithm uses 33.33%, 10%, 5%, 4.16%, 30.30%, 33.33% and 21.12% function evaluations to obtain competitive results compared to the $(\mu+\lambda)$-evolutionary strategy [$(\mu+\lambda)$-ES], UPSO, CPSO, CoDE, PSO-DE, ABC and MBA, respectively.

For the pressure vessel (problem 7), the Jaya algorithm is better than the other 10 algorithms, except for the Elitist TLBO algorithm, using fewer function evaluations to obtain the global optimum solution. The Elitist TLBO is superior to the other nine optimization algorithms and inferior to DETPS, but it requires far fewer function evaluation than DETPS. In terms of standard deviation, PSO-DE is superior and shows robustness. However, the Jaya algorithm uses 33.33%, 10%, 5%, 4.16%, 23.75%, 33.33% and 33.33% function evaluations to obtain competitive results compared to $(\mu+\lambda)$-ES, UPSO, CPSO, CoDE, PSO-DE, ABC and MBA, respectively.

For the tension/compression spring (problem 8), the Jaya algorithm is superior to the other 10 algorithms in terms of quality of solution, requiring fewer function evaluations to obtain the best and mean solutions and less computational time and effort. The Elitist TLBO is superior to UPSO and ABC and is inferior to DETPS, $(\mu+\lambda)$-ES, CPSO, CoDE, MBA and PSO-DE in terms of standard deviation. PSO-DE is the best optimization algorithm in terms of standard deviation for this problem.

For the speed reducer (problem 9), the Jaya algorithm produces the same results as the Elitist TLBO, DETPS, TLBO, $(\mu+\lambda)$-ES and PSO-DE, while MBA has the first rank in terms of the best solution among the algorithms. The Jaya algorithm, Elitist TLBO, DETPS, $(\mu+\lambda)$-ES and PSO-DE produce the same worst value. However, the Jaya algorithm, DETPS and Elitist TLBO need fewer function evaluations than $(\mu+\lambda)$-ES and PSO-DE to produce the same worst value. The Jaya algorithm, $(\mu+\lambda)$-ES and ABC are better and more robust optimization algorithms for this problem since zero standard deviation is achieved using these algorithms. The Elitist TLBO is inferior to the Jaya algorithm, $(\mu+\lambda)$-ES, PSO-DE and ABC in terms of standard deviation for this problem.

Table 3 presents the NFE for test problems 10–21 over 30 independent runs. The statistical results of the mean best solutions and standard deviations for problems 10–21 are shown in Table 4. For problem 10, the Jaya algorithm gives better solutions than the Elitist TLBO, DETPS, modified differential evolution (MDE), modified differential evolution–local search hybrid (MA-MDE) and modified differential evolution–improved harmony search hybrid (MDE-IHS). DETPS is better than the Jaya algorithm, Elitist TLBO, MDE, MA-MDE and MDE-IHS algorithms in terms of standard deviation. The first rank is obtained by the Jaya algorithm among these six algorithms and has better robustness than the Elitist TLBO, MDE, MA-MDE and MDE-IHS for problem 11. The experimental results in Table 4 show that the Jaya and Elitist TLBO algorithms achieve better solutions than the other four algorithms for problem 12. Also, it can be seen from Table 3 that the Jaya algorithm requires far fewer function evaluations than the other algorithms. For problem 13, the Jaya algorithm, Elitist TLBO, DETPS and MA-MDE achieve better solutions than the remaining algorithms considered for comparison. However, to achieve the best quality solution the Jaya algorithm requires fewer function evaluations than the other algorithms. The Jaya algorithm, Elitist TLBO, DETPS and MA-MDE show equal robustness in terms of the standard deviation.

For problem 14, the MA-MDE algorithm produces better solutions than the other optimization algorithms considered. For problem 15, the Jaya and Elitist TLBO algorithm provide better solutions, in terms of mean best solution, than the other algorithms. Moreover, the Jaya algorithm requires fewer function evaluations to achieve the global optimum. The Jaya algorithm is superior in terms of standard deviation and shows robustness.

For problem 16, the Jaya algorithm is competitive with the other five algorithms with respect to the quality of the solution and NFE. The Elitist TLBO algorithm is superior to the Jaya algorithm, DETPS, MDE, MA-MDE and MDE-IHS. However, to achieve the best mean solution the Elitist TLBO requires more function evaluations than the Jaya algorithm. DETPS achieves the best result in terms of standard deviation.

For problem 17, the Jaya and Elitist TLBO algorithms are superior to the other four optimization algorithms. The Jaya algorithm, DETPS and MDE-IHS have better robustness in terms of standard deviation.

For problem 18, the Jaya, Elitist TLBO, DETPS and MDE-IHS algorithms provide the same global optimum solution and are superior to the other two algorithms. However, to achieve the global optimum solution the Jaya algorithm requires fewer function evaluations than the Elitist TLBO, DETPS and MDE-HIS algorithms. DETPS and MDE-IHS are superior in terms of standard deviation.

For problem 19, the Jaya, Elitist TLBO, DETPS and MDE-IHS algorithms provide the same global optimum solution and are superior to the other two algorithms. However, to achieve the global optimum solution the Jaya algorithm requires fewer function evaluations than the Elitist TLBO, DETPS and MDE-HIS algorithms. DETPS and MDE-IHS are superior in terms of standard deviation.

For problem 20, DETPS produces better solutions than the other optimization algorithms but at the cost of more function evaluations. For problem 21, DETPS provides better solutions than the five other algorithms in terms of mean best solution and standard deviation. From Table 3 it can be observed that for problems 10–21 the NFE required to achieve the global optimum solution by the Jaya algorithm is lower than for the other five algorithms for all the problems considered for comparison.

The time taken by the Jaya algorithm for 30 runs in problems 1–21 is presented in Table 6. The results of Wilcoxon's test are presented in Table 7, which gives p values stating whether a statistical difference is significant or not. The smaller the p value, greater the difference between the considered algorithms. The Jaya algorithm shows similar performance to Elitist TLBO, DETPS, TLBO and MDE-IHS, and shows improvement over ABC, MDE and MA-MDE, with a level of significance (α) of 0.05.

Table 5 presents the comparison of results for the four mechanical design problems, *i.e.* robot gripper, multiple disc clutch brake, hydrostatic thrust bearing and rolling element bearing, obtained using the Jaya, TLBO and ABC algorithms. It is observed from Table 5 that for the robot gripper problem, the mean value obtained using the Jaya algorithm is better than the results obtained by the TLBO and ABC. The result of the Jaya algorithm is similar to the results of the TLBO and ABC algorithms in terms of the best solution. The Jaya algorithm is superior to the TLBO and ABC algorithms in terms of success rate (SR) for the robot gripper problem.

For the multiple disc clutch brake problem, the Jaya algorithm obtains first rank in terms of the mean solution among the three algorithms. The performance of the Jaya, TLBO and ABC algorithms

Table 6. Time taken by the Jaya algorithm for different experiments.

Problem	Time (s)
1	2.16
2	8.97
3	9.75
4	10.56
5	21.89
6	1.39
7	1.10
8	1.11
9	1.56
10	1.59
11	8.89
12	1.37
13	0.51
14	0.61
15	4.56
16	5.18
17	0.49
18	5.28
19	0.27
20	5.23
21	1.03

Table 7. Results of Wilcoxon's test.

Comparison	*p* value
Jaya versus DETPS	0.3928
Jaya versus TLBO	0.1241
Jaya versus ES	0.0619
Jaya versus ABC	0.0302
Jaya versus MDE	0.0009
Jaya versus MA-MDE	0.0051
Jaya versus MDE-IHS	0.1032
Jaya versus Elitist TLBO	0.3952

Note: DETPS = differential evolution algorithm and tissue P systems; TLBO = teaching–learning-based optimization; ES = evolution strategy; ABC = artificial bee colony; MDE = modified differential evolution; MA-MDE = modified differential evolution–local search hybrid; MDE-IHS = modified differential evolution–improved harmony search hybrid; Elitist TLBO = elitist teaching–learning-based optimization.

is similar in terms of the best solution. Also, it is observed from Table 5 that the SR obtained using the Jaya algorithm is better than for the other algorithms considered for comparison.

For the hydrostatic thrust bearing problem, the Jaya algorithm obtains first rank in terms of the mean solution among the three algorithms. The performance of the Jaya, TLBO and ABC algorithms is similar in terms of the best solution. Also, it is observed from Table 5 that the SR obtained using the Jaya algorithm is better than for the other algorithms considered for comparison.

For the rolling element bearing problem, the Jaya algorithm obtains first rank in terms of the best solution among the three algorithms. The performance of the ABC algorithm is superior to the Jaya and TLBO algorithms in terms of the mean solution, while the Jaya algorithm is superior to the TLBO and ABC algorithms in terms of the worst solution. Also, it is observed from Table 5 that the SR obtained using the Jaya algorithm is better than for the other algorithms considered for comparison.

5. Conclusions

In the present work, the performance of a very recently proposed Jaya algorithm is investigated on 21 well-defined constrained design optimization problems and four constrained mechanical design problems. For the considered optimization problems, comparisons are made between the results obtained using the Jaya algorithm and the other optimization algorithms. In addition, Wilcoxon's test is performed to check the statistical significance of the differences. The computational results reveal that the Jaya algorithm is competitive with or superior to the other optimization algorithms for the considered problems. More complex problems can be investigated using the Jaya algorithm in the near future. Various methods for adapting the common control parameters and constraint handling methods and their effects will also be researched.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

Ahrari, A., and A. A. Atai. 2010. "Grenade Explosion Method—A Novel Tool for Optimization of Multimodal Functions." *Applied Soft Computing* 10 (4): 1132–1140.
Akay, B., and D. Karaboga. 2010. "Artificial Bee Colony Algorithm for Large-Scale Problems and Engineering Design Optimization." *Journal of Intelligent Manufacturing*. doi:10.1007/s10845-010-0393-4.
Becerra, R. L., and C. A. C. Coello. 2006. "Cultured Differential Evolution for Constrained Optimization." *Computer Methods in Applied Mechanics and Engineering* 195: 4303–4322.
Blum, C. 2005. "Ant Colony Optimization: Introduction and Recent Trends." *Physics of Life Reviews* 2: 353–373.
Brajevic, I., and Tuba, M. 2013. "An Upgraded Artificial Bee Colony (ABC) Algorithm for Constrained Optimization Problems." *Journal of Intelligent Manufacturing* 24: 729–740.
Clerc, M. 2006. *Particle Swarm Optimization*. London: ISTE Publishing Company.
Deb, K. 2000. "An Efficient Constraint Handling Method for Genetic Algorithm." *Computer Methods in Applied Mechanics and Engineering* 186: 311–338.
Dorigo, M., and T. Stutzle. 2004. *Ant Colony Optimization*. Cambridge, MA: MIT Press.
Goldberg, D. 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*. New York: Addison-Wesley.
He, Q., and L. Wang. 2007. "An Effective Co-evolutionary Particle Swarm Optimization for Constrained Engineering Design Problems." *Engineering Applications of Artificial Intelligence* 20: 89–99.
Huang, F. Z., L. Wang, and Q. He. 2007. "An Effective Co-evolutionary Differential Evolution for Constrained Optimization." *Applied Mathematics and Computation* 186: 340–356.
Karaboga, D. 2005. "An Idea Based on Honey Bee Swarm for Numerical Optimization". Technical Report-TR06. Erciyes University, Engineering Faculty, Computer Engineering Department, Kayseri, Turkey.
Karaboga, D., and B. Basturk. 2007. "Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems." In *LNAI (IFSA)*, edited by P. Melin, Vol. 4529, 789–798. Berlin: Springer.
Kennedy, V., and R. Eberhart. 1995. "Particle Swarm Optimization." In *Proceedings of the IEEE International Conference on Neural* Networks, 1942–1948.
Lee, K. S., and Z. W. Geem. 2004. "A New Meta-heuristic Algorithm for Continuous Engineering Optimization: Harmony Search Theory and Practice." *Computer Methods in Applied Mechanics and Engineering* 194: 3902–3933.
Liao, T. W. 2010. "Two Hybrid Differential Evolution Algorithms for Engineering Design Optimization." *Applied Soft Computing* 10: 1188–1199.
Lin, M.-H., J.-F. Tsai, and C.-S. Yu. 2012. "A Review of Deterministic Optimization Methods in Engineering and Management." *Mathematical Problems in Engineering* 2012. doi:10.1155/2012/756023.
Liu, H., Z. Cai, and Y. Wang. 2010. "Hybridizing Particle Swarm Optimization with Differential Evolution for Constrained Numerical and Engineering Optimization." *Applied Soft Computing* 10: 629–640.
Mezura-Montes, E., and C. A. C. Coello. 2005. "Useful Infeasible Solutions in Engineering Optimization with Evolutionary Algorithms." In *LNAI (MICAI2005)*, edited by A. Gelbukh, Vol. 3789, 652–662. Berlin: Springer.
Parsopoulos, K. E., and M. N. Vrahatis. 2005. "Unified Particle Swarm Optimization for Solving Constrained Engineering Optimization Problems." In *LNCS (ICNC)*, edited by L. Wang, Vol. 3612, 582–591. Berlin: Springer.
Pavone, M., G. Narzisi, and G. Nicosia. 2012. "Clonal Selection: An Immunological Algorithm for Global Optimization over Continuous Spaces." *Journal of Global Optimization* 53 (4): 769–808.
Rao, R. V. 2016. "Jaya: A Simple and New Optimization Algorithm for Solving Constrained and Unconstrained Optimization Problems." *International Journal of Industrial Engineering Computations* 7: 19–34.
Rao, R. V., V. J. Savsani, and D. P. Vakharia. 2011. "Teaching-Learning-Based Optimization: A Novel Method for Constrained Mechanical Design Optimization Problems." *Computer-Aided Design* 43 (3): 303–315.
Rao, R. V., and G. G. Waghmare. 2014. "Complex Constrained Design Optimization Using an Elitist Teaching-Learning-Based Optimization Algorithm." *International Journal of Metaheuristics* 3 (1): 81–102.

Rios, L. M., and N. V. Sahinidis. 2013. "Derivative-Free Optimization: A Review of Algorithms and Comparison of Software Implementations." *Journal of Global Optimization* 56 (3): 1247–1293.
Sadollah, A., A. Bahreininejad, H. Eskandar, and M. Hamdi. 2013. "Mine Blast Algorithm: A New Population Based Algorithm for Solving Constrained Engineering Optimization Problems." *Applied Soft Computing* 13 (5): 2592–2612.
Zavala, A. E. M., A. H. Aguirre, and E. R. V. Diharce. 2005. "Constrained Optimization via Evolutionary Particle Swarm Optimization Algorithm (PESO)." In *Proc. GECCO*, 209–216.
Zhang, G., J. Cheng, M. Gheorghe, and Q. Meng. 2013. "A Hybrid Approach Based on Differential Evolution and Tissue Membrane Systems for Solving Constrained Manufacturing Parameter Optimization Problems." *Applied Soft Computing* 13 (3): 1528–1542. doi:10.1016/j.asoc.2012.05.032.

Appendix 1. Details of the 21 constrained design benchmark optimization problems and four mechanical design problems

Constrained design benchmark optimization problems

Problem 1

$$\min f(x) = 5\sum_{i=1}^{4} x_i - 5\sum_{i=1}^{4} {x_i}^2 - \sum_{i=5}^{13} x_i \tag{A1}$$

$$\text{s.t. } g_1(x) = 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \le 0 \tag{A2}$$

$$g_2(x) = 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \le 0 \tag{A3}$$

$$g_3(x) = 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \le 0 \tag{A4}$$

$$g_4(x) = -8x_1 + x_{10} \le 0 \tag{A5}$$

$$g_5(x) = -8x_2 + x_{11} \le 0 \tag{A6}$$

$$g_6(x) = -8x_3 + x_{12} \le 0 \tag{A7}$$

$$g_7(x) = -2x_4 - x_5 + x_{10} \le 0 \tag{A8}$$

$$g_8(x) = -2x_6 - x_7 + x_{11} \le 0 \tag{A9}$$

$$g_9(x) = -2x_8 - x_9 + x_{12} \le 0 \tag{A10}$$

where $0 \le x_i \le 1, i = 1,2,3, \ldots, 9; 0 \le x_i \le 100, i = 10,11,12; 0 \le x_{13} \le 1$.
The optimal solution is $f(x^*) = -15$ at $x^* = (1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 1)$.

Problem 2

$$\max f(x) = \left(\sqrt{n}\right)^n \prod_{i=1}^{n} x_i \tag{A11}$$

$$\text{s.t. } h(x) = \sum_{i=1}^{n} {x_i}^2 - 1 = 0 \tag{A12}$$

where $n = 10$ and $0 \le x_i \le 10, i = 1,2,3, \ldots, n$.
The global maximum $f(x^*) = 1$ at $x^* = (1/n^{0.5}, 1/n^{0.5}, \ldots)$.

Problem 3

$$\min f(x) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + {x_3}^4 + 3(x_4 - 11)^2 + 10{x_5}^6 + 7{x_6}^2 + {x_7}^4 - 4x_6x_7 - 10x_6 - 8x_7 \tag{A13}$$

$$\text{s.t. } g_{1(x)} = -127 + 2{x_1}^2 + 3{x_2}^4 + x_3 + 4{x_4}^2 + 5x_5 \le 0 \tag{A14}$$

$$g_{2(x)} = -282 + 7x_1 + 3x_2 + 10{x_3}^2 + x_4 - x_5 \le 0 \tag{A15}$$

$$g_{3(x)} = -196 + 23x_1 + {x_2}^2 + 6{x_6}^2 - 8x_7 \le 0 \tag{A16}$$

$$g_{4(x)} = 4{x_1}^2 + {x_2}^2 - 3x_1x_2 + 2{x_3}^2 + 5x_6 - 11x_7 \le 0 \tag{A17}$$

where $-10 \le x_i \le 10, i = 1,2,3, \ldots, 7$.

The optimal solution is $f(x^*) = 680.6300573$ at $x^* =$ (2.330499, 1.951372, −0.4775414, 4.365726, −0.6244870, 1.1038131, 1.594227).

Problem 4

$$\min f(x) = x_1 + x_2 + x_3 \tag{A18}$$
$$\text{s.t. } g_{1(x)} = -1 + 0.0025(x_4 + x_6) \le 0 \tag{A19}$$
$$g_{2(x)} = -1 + 0.0025(x_5 + x_7 - x_4) \le 0 \tag{A20}$$
$$g_{3(x)} = -1 + 0.01(x_8 - x_5) \le 0 \tag{A21}$$
$$g_{4(x)} = -x_1x_6 + 833.3325x_4 + 100x_1 - 83333.333 \le 0 \tag{A22}$$
$$g_{5(x)} = -x_2x_7 + 1250x_5 + x_2x_4 - 1250x_4 \le 0 \tag{A23}$$
$$g_{6(x)} = -x_3x_8 + 1250000 + x_3x_5 - 2500x_5 \le 0 \tag{A24}$$

where $100 \le x_1 \le 10{,}000$, $1000 \le x_i \le 10{,}000$, $i = 2,3$, $100 \le x_i \le 10{,}000$, $i = 4,5,\ldots,8$.

The optimal solution is $f(x^*) = 7049.248021$ at $x^* =$ (579.3066, 1359.9709, 510.9707, 182.0177, 295.601, 217.982, 286.165, 395.6012).

Problem 5

$$\max f(x) = \frac{100 - (x_1 - 5)^2 - (x_2 - 5)^2 - (x_3 - 5)^2}{100} \tag{A25}$$
$$\text{s.t. } g(x) = (x_1 - p)^2 - (x2 - q)^2 - (x_3 - r)^2 \le 0 \tag{A26}$$

where $0 \le x_i \le 10$, $i = 1,2,3$, $p,q,r = 1,2,3,\ldots,9$.

The optimal solution is $f(x^*) = 1$ at $x^* =$ (5,5,5).

Problem 6

This is a welded beam design problem, which is designed for the minimum cost subject to constraints on shear stress (τ), bending stress in the beam (σ), buckling load on the bar (P_c), end deflection of the beam (δ) and side constraints. There are four design variables, *i.e.* $h(x_1)$, $l(x_2)$, $t(x_3)$ and $b(x_4)$. This problem can be mathematically formulated as follows:

$$\min f(x) = 1.10471{x_1}^2x_2 + 0.04811x_3x_4(14.0 + x_2) \tag{A27}$$
$$\text{s.t. } g_1(x) = \tau(x) - \tau_{\max} \le 0 \tag{A28}$$
$$g_2(x) = \sigma(x) - \sigma_{\max} \le 0 \tag{A29}$$
$$g_3(x) = x_1 - x_4 \le 0 \tag{A30}$$
$$g_4(x) = 0.10471{x_1}^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \le 0 \tag{A31}$$
$$g_5(x) = 0.125 - x_1 \le 0 \tag{A32}$$
$$g_6(x) = \delta(x) - \delta_{\max} \le 0 \tag{A33}$$
$$g_6(x) = P - P_c(x) \le 0 \tag{A34}$$

where

$$\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau')^2} \tag{A35}$$
$$\tau' = \frac{P}{2^{0.5}x_1x_2} \tag{A36}$$
$$\tau'' = \frac{\mathrm{MR}}{J} \tag{A37}$$
$$M = P\left(L + \frac{x_2}{2}\right) \tag{A38}$$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2} \tag{A39}$$

$$J = 2\left\{2^{0.5} x_1 x_2 \left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)\left(\frac{x_1 + x_3}{2}\right)\right]\right\} \tag{A40}$$

$$\sigma(x) = \frac{6PL}{x_4 x_3^2} \tag{A41}$$

$$\delta(x) = \frac{4PL^3}{E x_3^3 x_4} \tag{A42}$$

$$P_c(x) = \frac{4.013E\sqrt{x_3^2 x_4^6/36}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right) \tag{A43}$$

where $P = 6000$ lb, $L = 14$ in., $E = 30 \times 10^6$ psi, $G = 12 \times 10^6$ psi, $\tau_{\max} = 13{,}600$ psi, $\sigma_{\max} = 30{,}000$ psi, $\delta_{\max} = 0.25$ in., $0.1 \le x_1 \le 2$, $0.1 \le x_2 \le 10$, $0.1 \le x_3 \le 10$, $0.1 \le x_4 \le 2$.

Problem 7

This is a pressure vessel design problem for minimizing the total cost $f(x)$ of a pressure vessel considering the cost of the material, forming and welding. There are four design variables: *Ts* (x_1, thickness of the shell), *Th* (x_2, thickness of the head), R (x_3, inner radius) and L (x_4, length of the cylindrical section of the vessel, not including the head). Among the four variables, *Ts* and *Th*, which are integer multiples of 0.0625 in., are the available thickness of rolled steel plates, and R and L are continuous variables. This problem can be formulated as follows:

$$\min f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \tag{A44}$$

$$\text{s.t. } g_1(x) = -x_1 + 0.0193x_3 \le 0 \tag{A45}$$

$$g_2(x) = -x_2 + 0.00954x_3 \le 0 \tag{A46}$$

$$g_3(x) = -\Pi x_3^2 x_4 - \frac{4}{3}\Pi x_3^2 + 1296000 \le 0 \tag{A47}$$

$$g_4(x) = x_4 - 240 \le 0 \tag{A48}$$

where

$$1 \le x_1 \le 99, 1 \le x_2 \le 99, 10 \le x_3 \le 200, 10 \le x_4 \le 200 \tag{A49}$$

Problem 8

This is a tension/compression spring design problem for minimizing the weight ($f(x)$) of a tension/compression spring subject to constraints on minimum deflection, shear stress, surge frequency, and limits on outside diameter and on design variables. The design variables are the mean coil diameter D (x_2), the wire diameter d (x_1) and the number of active coils P (x_3). The mathematical formulation of this problem can be described as follows:

$$\min f(x) = (x_3 + 2)x_2x_1^2 \tag{A50}$$

$$\text{s.t. } g_1(x) = 1 - \frac{x_2^3 x_3}{71785x_1^4} \le 0 \tag{A51}$$

$$g_2(x) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \le 0 \tag{A52}$$

$$g_3(x) = 1 - \frac{140.45x_1}{x_2^2 x_3} \le 0 \tag{A53}$$

$$g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \le 0 \tag{A54}$$

where $0.05 \le x_1 \le 2$, $0.25 \le x_2 \le 1.3$, $2 \le x_3 \le 15$.

Problem 9

This is a speed reducer design problem for minimizing the weights of the speed reducer subject to constraints on bending stress of the gear teeth, surface stress, transverse deflections of the shafts and stresses in the shafts. The parameters $x_1, x_2, \ldots, x_7$ represent the face width (b), module of the teeth (m), number of teeth in the pinion (z), length of the first shaft between bearings (l_1), length of the second shaft between bearings (l_2), and the diameters of the first shaft (d_1) and the second shaft (d_2).

$$\min f(x) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3) \quad \text{(A55)}$$

$$\text{s.t. } g_1(x) = \frac{27}{x_1x_2^2x_3} - 1 \leq 0 \quad \text{(A56)}$$

$$g_2(x) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \leq 0 \quad \text{(A57)}$$

$$g_3(x) = \frac{1.93x_4^3}{x_2x_3x_6^4} - 1 \leq 0 \quad \text{(A58)}$$

$$g_4(x) = \frac{1.93x_5^3}{x_2x_3x_7^4} - 1 \leq 0 \quad \text{(A59)}$$

$$g_5(x) = \frac{\sqrt{(745_{x_4}/x_2x_3)^2 + 16.9 \times 10^6}}{110.0x_6^3} - 1 \leq 0 \quad \text{(A60)}$$

$$g_6(x) = \frac{\sqrt{(745_{x_4}/x_2x_3)^2 + 157.5 \times 10^6}}{85.0x_6^3} - 1 \leq 0 \quad \text{(A61)}$$

$$g_7(x) = \frac{x_2x_3}{40} - 1 \leq 0 \quad \text{(A62)}$$

$$g_8(x) = \frac{5x_2}{x_1} - 1 \leq 0 \quad \text{(A63)}$$

$$g_9(x) = \frac{x_1}{12x_2} - 1 \leq 0 \quad \text{(A64)}$$

$$g_{10}(x) = \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0 \quad \text{(A65)}$$

$$g_{11}(x) = \frac{1.1x_7 + 1.9}{x_5} - 1 \leq 0 \quad \text{(A66)}$$

where $2.6 \leq x_1 \leq 3.6$, $0.7 \leq x_2 \leq 0.8$, $17 \leq x_3 \leq 28$, $7.3 \leq x_4 \leq 8.3$, $7.8 \leq x_5 \leq 8.3$, $2.9 \leq x_6 \leq 3.9$, $5.0 \leq x_7 \leq 5.5$.

Problem 10

$$\min f = 7.5y_1 + 6.4x_1 + 5.5y_2 + 6.0x_2 \quad \text{(A67)}$$

$$\text{s.t. } 0.8x_1 + 0.67x_2 = 10 \quad \text{(A68)}$$

$$x_1 - 20y_1 \leq 0 \quad \text{(A69)}$$

$$x_2 - 20y_1 \leq 0 \quad \text{(A70)}$$

where $x_1, x_2 \in [0,20]; y_1, y_2 \in 0,1\}$.

The global optimum f^* is 87.5 at $x =$ [12.5006, 0] and $y =$ [1, 0].

Problem 11

$$\min f = 2x_1 + 3x_2 + 1.5y_1 + 2y_2 - 0.5y_3 \quad \text{(A71)}$$

$$\text{s.t. } (x_1)^2 + y_1 = 1.25 \quad \text{(A72)}$$

$$(x_2)^{1.5} + 1.5y^2 = 3 \quad \text{(A73)}$$

$$x_1 + y_1 \leq 1.6 \quad \text{(A74)}$$

$$1.333x_2 + y_2 \leq 3 \quad \text{(A75)}$$

$$-y_1 - y_2 + y_3 \leq 0 \quad \text{(A76)}$$

where $x_1, x_2 \in [0, 2]; y_1, y_2, y_3 \in \{0, 1\}$.

The global optimum f^* is 7.667 at $x = [1.118, 1.310]$ and $y = [0, 1, 1]$.

Problem 12

$$\min f = (y_1 - 1)^2 + (y_2 - 2)^2 + (y_3 - 1)^2 - \ln(y_4 - 1) + (x_1 - 1)^2 + (x_2 - 2)^2 + (x_3 - 3)^2 \tag{A77}$$

$$\text{s.t. } y_1 + y_2 + y_3 + x_1 + x_2 + x_3 \le 5 \tag{A78}$$

$$y_3^2 + x_1^2 + x_2^2 + x_3^2 \le 5.5 \tag{A79}$$

$$y_1 + x_1 \le 1.2 \tag{A80}$$

$$y_2 + x_2 \le 1.8 \tag{A81}$$

$$y_3 + x_3 \le 2.5 \tag{A82}$$

$$y_4 + x_1 \le 1.2 \tag{A83}$$

$$y_2^2 + x_2^2 \le 1.64 \tag{A84}$$

$$y_3^2 + x_3^2 \le 1.25 \tag{A85}$$

$$y_2^2 + x_3^2 \le 4.64 \tag{A86}$$

where $x_1 \in [0, 1, 2]; x_2 \in [0, 1.281]; x_3 \in [0, 2.062]; y_1, y_2, y_3, y_4 \in \{0, 1\}$.

The global optimum $f^*$is 4.5796 at $x = [0.2, 0.8, 1.908]$ and $y = [1, 1, 0, 1]$.

Problem 13

This is a process synthesis problem.

$$\min f = 2x + y \tag{A87}$$

$$\text{s.t. } 1.25 - x^2 - y \le 0 \tag{A88}$$

$$x + y \le 1.6 \tag{A89}$$

where $x \in [0, 1.6]; y \in \{0, 1\}$.

The global optimum $f^*$is 2 at $x = 0.5$ and $y = 1$.

Problem 14

This is a process synthesis design problem.

$$\min f = y + 2x_1 - \ln\left(\frac{x_1}{2}\right) \tag{A90}$$

$$\text{s.t. } -x_1 - \ln\left(\frac{x_1}{2}\right) + y \le 0 \tag{A91}$$

where $x_1 \in [0.5, 1.4]; y \in \{0, 1\}$.

The global optimum $f^*$is 2.1247 at $x_1 = 1.375$ and $y = 1$.

Problem 15

This is a process flow sheeting problem.

$$\min f = -0.7y + 5(x_1 - 0.5)^2 + 0.8 \tag{A92}$$

$$\text{s.t. } -\exp(x_1 - 0.2) - x_2 \le 0 \tag{A93}$$

$$x_2 + 1.1y \le -1.0 \tag{A94}$$

$$x_1 - 1.2y \le 0.2 \tag{A95}$$

where $x_1 \in [0.2, 1]; x_2 \in [-2.22554, -1]; y \in \{0, 1\}$.

The global optimum $f^*$is 1.076543 at $x = [0.94194, 2.1]$ and $y = 1$.

Problem 16

This is a process synthesis problem.

$$\min f = (y_1-1)^2+(y_2-2)^2+(y_3-1)^2-\ln(y_4-1)+(x_1-1)^2+(x_2-2)^2+(x_3-3)^2 \tag{A96}$$

$$\text{s.t. } y_1+y_2+y_3+x_1+x_2+x_3 \le 5 \tag{A97}$$

$$y_3^2+x_1^2+x_2^2+x_3^2 \le 5.5 \tag{A98}$$

$$y_1+x_1 \le 1.2 \tag{A99}$$

$$y_2+x_2 \le 1.8 \tag{A100}$$

$$y_3+x_3 \le 2.5 \tag{A101}$$

$$y_4+x_1 \le 1.2 \tag{A102}$$

$$y_2^2+x_2^2 \le 1.64 \tag{A103}$$

$$y_3^2+x_3^2 \le 4.25 \tag{A104}$$

$$y_2^2+x_3^2 \le 4.64 \tag{A105}$$

where $x_1 \in [0,1.2]; x_2 \in [0,1.8]; x_3 \in [0,2.5]; y_1,y_2,y_3,y_4 \in \{0,1\}$.

The global optimum $f^*$is 3.557473 at $x=[0.2,1.28062,1.95448]$ and $y=[1,0,0,1]$.

Problem 17

This is a process design problem.

$$\min f = 5.357854x_1^2+0.835689y_1x_3+37.29329y_1-40792.141 \tag{A106}$$

$$\text{s.t. } 85.334407+0.0056858y_2x_3+0.0006262y_1x_2-0.0022053x_1x_3 \le 92 \tag{A107}$$

$$80.51249+0.0071317y_2x_3+0.0029955y_1y_2+0.0021813x_1^2-90 \le 20 \tag{A108}$$

$$9.300961+0.0047026x_1x_3+0.0012547y_1x_1+0.0019085x_1x_2-20 \le 5 \tag{A109}$$

where $x_1,x_2,x_3 \in [27,45]; y_1 \in \{78,\ldots,102\}$, integer; $y_2 \in \{33,\ldots,45\}$, integer.

The global optimum $f^*$is -32217.4 at $x=[27,\ \text{any}, 27]$ and $y=[78,\ \text{any}]$.

Problem 18

$$\min\ f = -\prod_{j=1}^{10}[1-(1-p_j)m_j] \tag{A110}$$

$$\text{s.t. } \prod_{j=1}^{10}(a_{ij}m_j^2+c_{ij}m_j) \le b_i, i=1,2,3,4 \tag{A111}$$

$$[p_j]=(0.81,0.93,0.92,0.96,0.99,0.89,0.85,0.83,0.94,0.92)$$

$$[a_{ij}]=\begin{bmatrix}2730569481\\4927108356\\5174360982\\8356972401\end{bmatrix}$$

$$[c_{ij}]=\begin{bmatrix}7146825933\\4657269108\\11035478946\\23257861091\end{bmatrix}$$

$$[b_i]=(2.0\times10^{13},3.1\times10^{12},5.7\times10^{13},9.3\times10^{12})$$

$m_j \in [1,6], j=1,\ldots,10$.

The global optimum f^* is -0.808844 at $m=$ [2,2,2,1,1,2,3,2,1,2].

Problem 19

$$\min f = -\prod_{j=1}^{4} R_j \tag{A112}$$

$$\text{s.t. } \sum_{j=1}^{4} d_{1j} m_j^2 \leq 100 \tag{A113}$$

$$\sum_{j=1}^{4} d_{2j} \left(m_j + \exp\left(\frac{m_j}{4}\right)\right) \leq 150 \tag{A114}$$

$$\sum_{j=1}^{4} d_{3j} m_j \exp\left(\frac{m_2}{4}\right) \leq 160 \tag{A115}$$

$$m_j \in [1, 6], j = 1, 2, 4$$
$$m_3 \in [1, 5]$$

where

$$R_1 = 1 - q_1((1-\beta_1)q_1 + \beta_1)^{m_1 - 1} \tag{A116}$$

$$R_2 = 1 - \frac{\beta_2 q_2 + p_2 q_2^{m_2}(1-\beta_2)^{m_2}}{p_2 + \beta_2 q_2} \tag{A117}$$

$$R_3 = 1 - q_3^{m_3} \tag{A118}$$

$$R_4 = 1 - q_4((1-\beta_4)q_4 + \beta_4)^{m_4 - 1} \tag{A119}$$

$$[p_j] = (0.93, 0.92, 0.94, 0.91)$$
$$[q_j] = (0.07, 0.08, 0.06, 0.09)$$
$$[\beta_j] = (0.2, 0.06, 0.0, 0.3)$$
$$[d_{ij}] = \begin{bmatrix} 1234 \\ 7757 \\ 7886 \end{bmatrix}$$

The global optimum f^* is -0.974565 at $m = [3, 3, 2, 3]$.

Problem 20

This is also a pressure vessel design problem for minimizing the total cost $f(x)$ of a pressure vessel considering the cost of material, forming and welding. A cylindrical vessel is capped at both ends by hemispherical heads. There are four design variables: R (x_1, inner radius), L (x_2, length of the cylindrical section of the vessel, not including the head), Ts (x_3, thickness of the shell) and Th (x_4, thickness of the head). This problem can be formulated as follows:

$$\min f = 0.6224 x_1 x_2 x_3 + 1.7781 x_1^2 x_4 + 3.1661 x_2 x_3^2 + 19.84 x_1 x_3^2 \tag{A120}$$

$$\text{s.t. } 0.0193 \frac{x_1}{x_3} - 1 = 0 \tag{A121}$$

$$0.00954 \frac{x_1}{x_4} - 1 = 0 \tag{A122}$$

$$\frac{x_2}{240} - 1 = 0 \tag{A123}$$

$$\frac{1296000 - \frac{4}{3}\pi x_1^3}{\pi x_1^2 x_2} - 1 \leq 0 \tag{A124}$$

where $x_1 \in [25, 150]$; $x_2 \in [25, 240]$; $x_3, x_4 \in [0.0625, 0.125, \ldots, 1.1875, 1.25]$.

The global optimum f^* is 5850.770 at $x =$ [38.858,221.402,0.750,0.375].

Problem 21

This is a ceramic grinding process optimization problem for maximizing the material removal rate subject to a set of constraints on surface roughness, number of flaws and input variables. The problem is formulated as

$$\min F = -f d_c \tag{A125}$$

$$\text{s.t. } 0.145 d_c^{0.1939} f^{0.7071} M^{-0.2343} = Ra_{(\max)} \tag{A126}$$

$$29.67 d_c^{0.4167} f^{-0.8333} = N_{c(\max)} \tag{A127}$$

where d_c is the depth of cut, $d_c \in [5,30]$ μm; f is the table feed rate, $f \in [8.6,13.4]$ m/min; $Ra_{(\max)}$ and $N_{c(\max)}$ are the maximum allowable values of surface roughness and number of flaws; and M is the grit size, $M \in \{120, 140, 170, 200, 230, 270, 325, 400, 500\}$.

The global optimum f^* is -75.1341 at $d_c = 5.6070$ and $f = 13.4$ when $Ra_{(\max)} = 0.3$ and $N_{c(\max)} = 7$.

Constrained mechanical design problems

1. Robot gripper

$$\text{Minimize } f(x) = \max_z F_k(x,z) - \min_z F_k(x,z) \tag{A128}$$

subject to:

$$g_1(x) = Y_{min} - y(x, Z_{\max}) \geq 0 \tag{A129}$$

$$g_2(x) = y(x, Z_{\max}) \geq 0 \tag{A130}$$

$$g_3(x) = y(x, 0) - Y_{\max} \geq 0 \tag{A131}$$

$$g_4(x) = Y_G - y(x, 0) \geq 0 \tag{A132}$$

$$g_5(x) = (a+b)^2 - l^2 - e^2 \geq 0 \tag{A133}$$

$$g_6(x) = (l - Z_{\max})^2 + (a-e)^2 - b^2 \geq 0 \tag{A134}$$

$$g_7(x) = l - Z_{\max} \geq 0 \tag{A135}$$

where

$$g = \sqrt{l - z^2 + e^2} \tag{A136}$$

$$\alpha = \arccos(a^2 + g^2 - b^2/2ag) + \phi \tag{A137}$$

$$\beta = \arccos(b^2 + g^2 - a^2/2bg) - \phi \tag{A138}$$

$$\phi = \arctan(e/l - z) + \phi \tag{A139}$$

$$F_k = (pb \sin(\alpha + \beta)/2c \cos(\alpha)) \tag{A140}$$

$$y(x,z) = 2(e + f + c \sin(\beta + \delta)) \tag{A141}$$

$Y_{\min} = 50$, $Y_{\max} = 100$, $Y_G = 150$, $Z_{\max}$=100, $P = 100$, $10 \leq a, b, f \leq 150$, $100 \leq c \leq 200$, $0 \leq e \leq 50$, $100 \leq l \leq 300$, $1 \leq \delta \leq 3.14$.

2. Multiple disc clutch brake design

$$\text{Min} f(x) = \pi(r_o^2 - r_i^2) t (Z+1) \rho \tag{A142}$$

subject to:

$$g_1(x) = r_o - r_i - \Delta r \geq 0 \tag{A143}$$

$$g_2(x) = l_{\max} - (Z+1)(t+\delta) \geq 0 \tag{A144}$$

$$g_3(x) = p_{\max} - p_{\text{rz}} \geq 0 \tag{A145}$$

$$g_4(x) = p_{\max} v_{\text{srmax}} - p_{\text{rz}} v_{\text{sr}} \geq 0 \tag{A146}$$

$$g_5(x) = v_{\text{srmax}} - v_{\text{sr}} \geq 0 \tag{A147}$$

$$g_6(x) = T_{\max} - T \geq 0 \tag{A148}$$

$$g_7(x) = M_h - sM_s \geq 0 \tag{A149}$$

$$g_8(x) = T \geq 0 \tag{A150}$$

where

$$M_h = \frac{2}{3}\mu FZ\frac{r_o^3 - r_i^3}{r_o^2 - r_i^2} \tag{A151}$$

$$p_{\rm rz} = \frac{F}{\pi(r_o^2 - r_i^2)} \tag{A152}$$

$$v_{\rm sr} = \frac{2\pi n(r_o^3 - r_i^3)}{90(r_o^2 - r_i^2)} \tag{A153}$$

$$T = \frac{I_z \pi n}{30(M_h + M_f)} \tag{A154}$$

$\Delta r = 20\,\text{mm}, t_{\max} = 3\,\text{mm}, \quad t_{\max} = 1.5\,\text{mm}, \quad \mu = 0.5, s = 1.5, M_s = 40\,\text{Nm}, M_f = 3\,\text{Nm}, n = 250\,\text{rpm}, p_{\max} = 1\,\text{MPa}, I_z = 55\,\text{kg mm}^2, T_{\max} = 15\text{s}, F_{\max} = 1000\,\text{N}, r_{i\min} = 55\,\text{mm}, r_{o\max} = 110\,\text{mm}.$

3. Hydrodynamic thrust bearing design

$$\text{Minimize}: f(x) = \frac{QP_o}{0.7} + E_f \tag{A155}$$

subject to:

$$g_1(x) = W - W_s \geq 0 \tag{A156}$$

$$g_2(x) = P_{\max} - P_o \geq 0 \tag{A157}$$

$$g_3(x) = \Delta T_{\max} - \Delta T \geq 0 \tag{A158}$$

$$g_4(x) = h - h_{\min} \geq 0 \tag{A159}$$

$$g_5(x) = R - R_o \geq 0 \tag{A160}$$

$$g_6(x) = 0.001 - \frac{\gamma}{gP_o}\left(\frac{Q}{2\pi Rh}\right) \geq 0 \tag{A161}$$

$$g_7(x) = 5000 - \frac{W}{\pi(R^2 - R_o^2)} \geq 0 \tag{A162}$$

where

$$W = \frac{\pi P_o}{2}\frac{R^2 - R_o^2}{lnR/R_o} \tag{A163}$$

$$P_o = \frac{6\mu Q}{\pi h^3} ln\frac{R}{R_o} \tag{A164}$$

$$E_f = 9336Q\gamma C\Delta T \tag{A165}$$

$$\Delta T = 2(10^P - 560) \tag{A166}$$

$$P = \frac{\log_{10}\log_{10}(8.122e6\mu + 0.8) - C_1}{n} \tag{A167}$$

$$h = \left(\frac{2\pi N}{60}\right)^2 \frac{2\pi\mu}{E_f}\left(\frac{R^4}{4} - \frac{R_o^4}{4}\right) \tag{A168}$$

where

$\gamma = 0.0307, C = 0.5, n = -3.55, C_1 = 10.04, W_s = 101,000, P_{\max} = 1000, \Delta T_{\max} = 50, h_{\min} = 0.001, g = 386.4, N = 750.$

$1 \leq R, R_o, Q \leq 16$, 1e-6 $\leq \mu \leq$ 16e-6.

4. Rolling element bearing

$$\text{Maximize}C_d = f_c Z^{2/3} D_b^{1.8} \text{ if } D_b \leq 25.4\,\text{mm} \tag{A169}$$

$$C_d = f_c Z^{2/3} D_b^{1.8} \text{ if } D_b \leq 25.4\,\text{mm} \tag{A170}$$

subject to:

$$g_1(x) = \frac{\phi_o}{2\sin^{-1}(D_b/D_m)} - Z + 1 \geq 0 \quad \text{(A171)}$$

$$g_2(x) = 2D_b - K_{D\min}(D - d) \geq 0 \quad \text{(A172)}$$

$$g_3(x) = K_{D\max}(D - d) - 2D_b \geq 0 \quad \text{(A173)}$$

$$g_4(x) = \xi B_w - D_b \geq 0 \quad \text{(A174)}$$

$$g_5(x) = D_m - 0.5(D + d) \geq 0 \quad \text{(A175)}$$

$$g_6(x) = (0.5 + e)(D + d) - D_m \geq 0 \quad \text{(A176)}$$

$$g_7(x) = 0.5(D - D_m - D_b) - \xi D_b \geq 0 \quad \text{(A177)}$$

$$g_8(x) = f_1 \geq 0.515 \quad \text{(A178)}$$

$$g_9(x) = f_o \geq 0.515 \quad \text{(A179)}$$

where

$$f_c = 37.91\left[1 + \left\{1.04\left(\frac{1-\gamma}{1+\gamma}\right)^{1.72}\left(\frac{f_1(2f_o - 1)}{f_o(2f_1 - 1)}\right)^{0.41}\right\}^{10/3}\right]^{-0.3} \quad \text{(A180)}$$

$$\gamma = \frac{D_b \cos\alpha}{D_m} \quad \text{(A181)}$$

$$f_1 = \frac{r_1}{D_b} \quad \text{(A182)}$$

$$\phi_o = 2\pi - 20\cos^{-1}\frac{[\{(D-d)/2 - 3(T/4)^2 + \{D/2 - (T/4 - D_b)\}^2 - \{d/2 + (T/4)\}^2\}]}{2\{D - d/2 - 3(T/4)\}\{D/2 - (T/4) - D_b\}} \quad \text{(A183)}$$

$$T = D - d - 2D_b \quad \text{(A184)}$$

$$D = 160, d = 90, B_w = 30 \quad \text{(A185)}$$

$$0.5(D + d) \leq D_m \leq 0.6(D + d) \quad \text{(A186)}$$

$$0.15(D - d) \leq D_b \leq 0.45(D - d) \quad \text{(A187)}$$

$$4 \leq Z \leq 50$$

$$0.515 \leq f_1 \leq 0.6$$

$$0.515 \leq f_o \leq 0.6$$

$$0.4 \leq K_{D\min} \leq 0.5$$

$$0.6 \leq K_{D\max} \leq 0.7$$

$$0.3 \leq \epsilon \leq 0.4,$$

$$0.02 \leq e \leq 0.1$$

$$0.6 \leq \xi \leq 0.85$$