Applied Mathematical Modelling 40 (2016) 3951–3978

Contents lists available at ScienceDirect

Applied Mathematical Modelling

journal homepage: www.elsevier.com/locate/apm

Passing vehicle search (PVS): A novel metaheuristic algorithm

Poonam Savsani, Vimal Savsani*

Pandit Deendayal Petroleum University, Gandinagar, Gujarat, India

ARTICLE INFO

Article history:
Received 4 February 2015
Revised 6 October 2015
Accepted 21 October 2015
Available online 10 November 2015

Keywords:
Constrained optimization
Engineering design optimization
Meta-heuristics
Passing vehicle search (PVS) algorithm

ABSTRACT

Vehicle passing mechanisms on two-lane highways have been studied since the first decade in the twentieth century and many mathematical models have been proposed. In general, we learn from our surroundings and the experience of driving a vehicle on roads is not an exception. This motivated us to apply the mechanism of vehicle passing on a two-lane highway to optimization problems. Thus, we propose a new metaheuristic optimization algorithm called "passing vehicle search (PVS)," which considers the mathematics of vehicle passing on a two-lane highway. Similar to other metaheuristic methods, PVS is a population-based method that requires an initial set of solutions to start and it then searches for the optimum solution by following the mathematical characteristics of vehicles overtaking on a two-lane highway. A simplified mathematical model is developed for vehicles moving on two-lane highways, which is then modified to solve different optimization problems. We investigated the performance of PVS with various challenging engineering design optimization problems. The results demonstrated the superior effectiveness of PVS compared with other metaheuristic optimization algorithms.

1. Introduction

Optimization methods are broadly classified into deterministic methods and meta-heuristic methods. Deterministic methods are classical methods such as gradient-based methods, sequential quadratic programming (SQP), geometric programming, and the sub-space trust region method, which apply specific mathematical rules at each iteration and they always obtain the same solution under specified conditions. These types of methods may require gradient information, Hessian matrix information, and an initial starting point. The effectiveness of the solution depends on the starting point selected. The main drawbacks of these classical methods are that they can only find the local optimum solution (and thus they fail with multi-modal problems), which is ineffective for highly constrained problems as well as being difficult to code. However, many of these methods are available in the optimization toolbox of Matlab, which can be readily used to find the local optimum solutions. Due to these limitations,

Abbreviations: ABC, Artificial bee colony; ACO, Ant colony optimization; BA, Bat algorithm; BBO, Biogeography-based optimization; BIANCA, Multi-population GA; BV, Back vehicle; CSA, Cuckoo search algorithm; CVI-PSO, Constraint violation with interval arithmetic PSO; DE, Differential evolution; DEC-PSO, Diversity-enhanced constrained PSO; DELC, Differential evolution level comparison; DV, Design variable; FE, Function evaluations; FFA, Firefly algorithm; FPA, Flower pollinating algorithm; FS, Feasible solution; FV, Front vehicle; GA, Genetic algorithm; GADS, Genetic algorithm and direct search; GSA, Gravitation search algorithm; ISA, Interior search algorithm; LB, Lower bound; MBA, Mine blast algorithm; MDDE, Multi-member diversity-based DE; NC, Number of constraints; NFS, No feasible solution; NM–PSO, Nelder–Mead particle swarm optimization; NOG, Number of generations; OV, Oncoming vehicle; PS, Population size; PSO, Particle swarm optimization; PSO–DE, Particle swarm optimization-differential evolution; PVS, Passing vehicle search; SQP, Sequential quadratic programming; TLBO, Teaching-learning based optimization; UB, Upper bound; WCA, Water cycle algorithm.

* Corresponding author. Tel.: +91 79 2327 5484; fax: +91 79 2327 5030.
E-mail address: vimal.savsani@gmail.com, vimal.s@sot.pdpu.ac.in (V. Savsani).

http://dx.doi.org/10.1016/j.apm.2015.10.040
S0307-904X(15)00702-7/

classical methods are rarely used for practical engineering optimization problems at present. However, these methods are faster at finding the solution compared with meta-heuristic methods.

In last 30 years, many new meta-heuristic optimization algorithms have been developed and applied to diverse real-life problems. These methods have many names but the same purpose, such as population-based methods, heuristic methods, meta-heuristic methods, advanced optimization techniques, computational intelligence algorithms, non-traditional optimization techniques, and clever algorithms [1]. It is logical to classify genetic algorithms (GAs) [2], ant colony optimization (ACO) [3], particle swarm optimization (PSO) [4], and artificial immune algorithms [5] as traditional meta-heuristics because they are well-recognized algorithms and many books, research papers, and technical articles tend to do the same [1]. Furthermore, the algorithms developed after 1995 can be referred to as non-traditional meta-heuristic optimization algorithms, which include most of the algorithms developed in the last decade due to differences in their behavior. These algorithms have been studied widely based on modifications or applications of these algorithms. It is very difficult to mention all of these algorithms but we provide a broad classification for some. Most of the algorithms are nature-inspired based on different behaviors of organisms. In addition, some of these algorithms are designed according to the basic physics of nature or based on human-related activities. Thus, these algorithms can be broadly classified as: (a) animal-based algorithms, (b) plant-based algorithms, (c) physics-based algorithms, and (d) human activity-based algorithms. The animal-based algorithms include bee-inspired algorithms [6], biogeography-based optimization (BBO) algorithms [7], bacteria-inspired algorithms [8], bat-inspired algorithm (BA; [9]), cat optimization algorithms [10], cuckoo search algorithm (CSA; [11]), luminous insect-based algorithms [12,13], fish-inspired algorithms [14,15], frog-based algorithms [16], rat-inspired algorithms [17], cockroach-inspired algorithms [18], dove-based algorithms [19], eagle-based algorithms [20], goose-based algorithms [21], monkey search algorithms [22], and wolf colony-inspired algorithms [23]. Very few plant-based optimization algorithms have been developed but they include the invasive weed optimization algorithm [24] and flower-pollinating algorithm (FPA; [25]). Physics-based algorithms include the charged system search algorithm [26], electromagnetism-based algorithm [27], gravitational search algorithm [28], water drop algorithm [29], and water cycle algorithm (WCA; [30]). Some of the algorithms based on the mathematics of human activities include music-inspired algorithms [31], the imperialist competitive algorithm [32], harmony element algorithm [33,34], grenade explosion algorithm [35], and teaching-learning based optimization (TLBO; [36]). There have been numerous applications of these algorithms in many areas of research, including communication optimization, data mining, image processing, power system optimization, robot control, scheduling, engineering design optimization, stock market prediction, composite structures, expert system design, water resource management, cloud computing, inventory management, supply chain management, and structure optimization. However, as suggested by the "No Free Lunch" theorem [37], no single algorithm is suitable for all optimization problems. Thus, more research is required to investigate different algorithms, modify existing algorithms [38–40], and develop new algorithms to suit real-life applications with high efficiency.

In this study, we propose a new meta-heuristic optimization algorithm called "passing vehicle search" (PVS) based on the passing behavior of a vehicle moving on a two-lane highway. This algorithm can be characterized as a human activity-based algorithm. Like other meta-heuristics, the objective of the proposed algorithm is to find the global solution or near-optimal solutions for a given function. We tested the performance of the proposed algorithm with several challenging engineering design problems. The performance of PVS was checked based on the tendency of the algorithm to find the best solutions in different runs, the mean of the solutions obtained in different runs, computational effort, computational time, convergence, the Friedman rank test [41], and Holm–Sidak multiple comparisons test [42].

In the following sections, we discuss the mathematical modeling of passing vehicles, the steps required to implement PVS for optimization, the performance of PVS with various engineering design problems and different constrained benchmark problems, and finally we give our conclusions.

2. Mathematical modeling of passing vehicles on two-lane highways

In the most developed and developing countries, two-lane highways are common in major road networks. According to recent studies, two-lane highways comprise over 65% of the total urban and rural routes in the USA and about 54% in India. In addition, over 60% of accidents occur on rural two-lane highways [43]. The most important consideration during two-lane vehicle passing is identifying safe overtaking opportunities (passing), which depend on many complex inter-related parameters such as the availability of gaps in the opposing traffic stream, the speed and acceleration characteristics of individual vehicles, the characteristics of the traffic and driver, and road and weather conditions. Many studies have reported traffic simulations of two-lane highways that consider various factors such as overtaking, platoons, delays, traffic quality, and speed distribution [44–54,78,79].

Passing on a two-lane highway is achieved by using the lane reserved for oncoming vehicles, which depends on the passing demand and supply at a particular time and location. Passing is determined by speed differences between the vehicles moving on the road, where it occurs when there is sufficient headway relative to an oncoming vehicle and sufficient observable distance for the overtaking vehicle. Passing also depends on the driver's perspective because all drivers have different preferred speeds, which also depend on factors such as the type of journey, type of vehicle used, weather conditions, road geometry, traffic conditions, and the driver's personality [51,55]. Three major circumstances may occur when a vehicle approaches a slower vehicle: (a) it passes the slower vehicle "on the fly," (b) it follows a slower vehicle until a passing opportunity occurs, or (c) it follows a slower vehicle without the intention of passing. If vehicles do not pass a slower vehicle, platoons begin to form. Platoons also affect the preferred speed of vehicles. Thus, mathematical theories of two-lane traffic are difficult to establish and analyze [56].

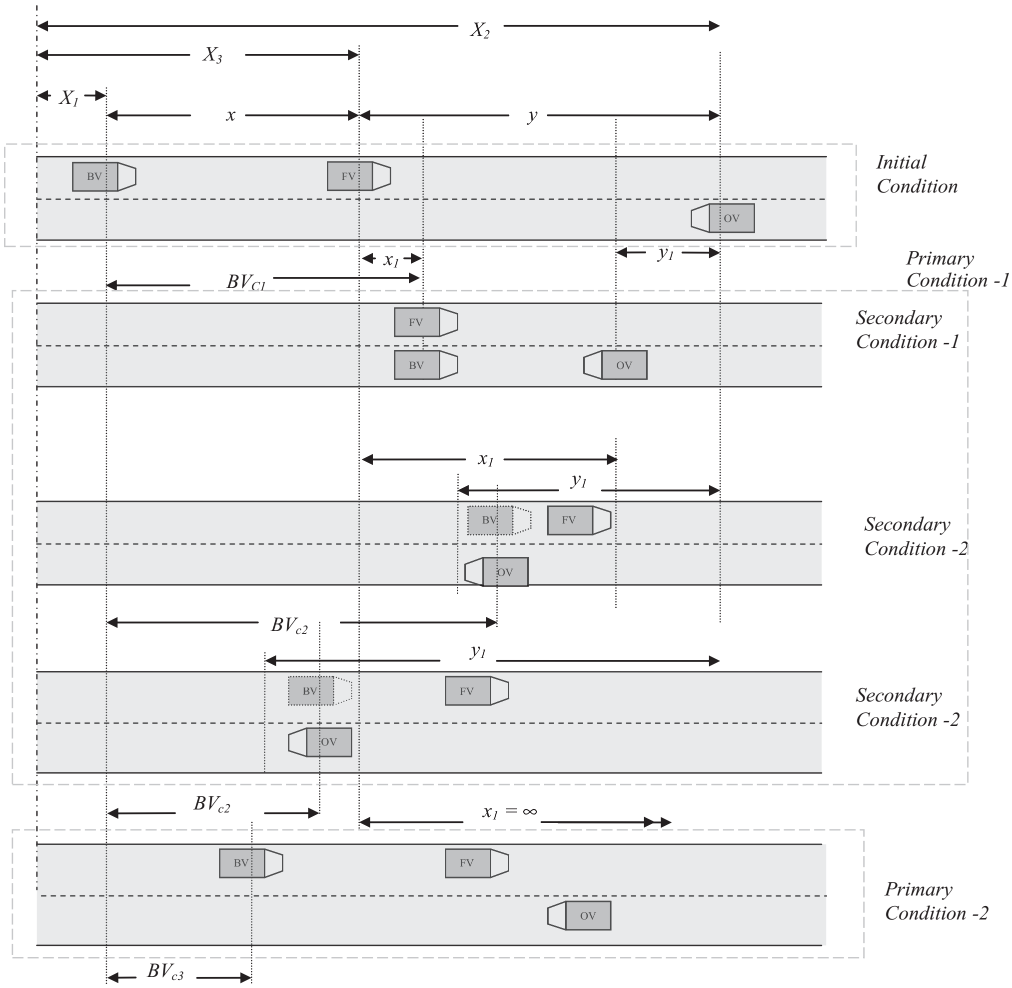

Fig. 1. Passing mechanism for a vehicle on a two-lane highway.

In this study, we formulate a simplified mathematical model for passing vehicles on two-lane highways. We consider three vehicles that are involved in the passing mechanism on a two-lane highway, as shown in Fig. 1: back vehicle (BV), front vehicle (FV), and oncoming vehicle (OV). Passing only occurs if the passing supply is more than the passing demand. BV wants to pass FV, which is only possible if FV is slower than BV. If FV is faster than BV, then passing is not possible by BV. Moreover, passing depends on the position and speed of OV as well as the distance between BV, FV, and OV, and their velocities, which generate different conditions, as discussed in the following.

Let,

x – Distance between BV and FV
y – Distance between FV and OV
X_1, X_2, X_3 – Distance from reference line
V_1, V_2, V_3 – Velocities of BV, OV and FV, respectively,

We assume that there are three different vehicles (*BV*, *FV* and *OV*) on a two-lane highway, which have different velocities (V_1, V_2 and V_3) at any particular time instance. Two primary conditions occur based on the velocity of *FV*, i.e., *FV* is slower than *BV* ($V_1 > V_3$) and vice versa. If *FV* is faster than *BV*, then passing is not possible and *BV* can move at its desired velocity. Passing is possible only if *FV* is slower than *BV*. In this situation, overtaking is only possible if the distance from *FV* at which overtaking occurs (x_1) is less than the distance travelled by *OV* ($y - y_1$). Thus, the following conditions are possible for the selected vehicles.

1. FV is slower than BV (V3 < V1) (Primary condition-1)
   (a) $(y - y1) > \text{x1}$ (Secondary condition-1)
   (b) $(y - y1) < \text{x1}$ (Secondary condition-2)
2. FV is faster than BV (V3 > V1) (Primary condition-2)

2.1. Primary condition-1

This condition deals with the case where FV is slower than BV, which is again divided into two sub-conditions: Secondary condition-1 and Secondary condition-2. Mathematical expressions for these conditions are given as follows.

2.1.1. Secondary condition-1

Let x_1 be the distance travelled by *FV* when *BV* catches *FV* and thus passes it. We consider that time is equal to "t" for *BV* to catch *FV*.

Thus, the distance travelled by *FV* in time t is given by:

$$x_1 = V_3 t. \tag{1}$$

The distance travelled by *BV* in time t is

$$x + x_1 = V_1 t. \tag{2}$$

Equating (1) and (2), we have

$$\frac{x_1}{V_3} = \frac{x + x_1}{V_1} \tag{3}$$

and thus,

$$x_1 = \frac{V_3 x}{V_1 - V_3}. \tag{4}$$

Now, the distance travelled by *OV* in time "t" is

$$y_1 = V_2 t. \tag{5}$$

By substituting the value of x_1 in Eq. (1), we obtain

$$t = \frac{x}{V_1 - V_3}. \tag{6}$$

If we substitute (6) in (5), we have

$$y_1 = \frac{V_2 x}{V_1 - V_3}. \tag{7}$$

Now, the change in the position of *BV* is given as follows.

$$BV_{c1} = x + x_1. \tag{8}$$

Substitute the value of x_1 from Eq. (4)

$$BV_{c1} = x\left(\frac{V_1}{V_1 - V_3}\right). \tag{9}$$

Substitute the value of x that corresponds to the reference line, which leads to:

$$BV_{c1} = (X_3 - X_1)\left(\frac{V_1}{V_1 - V_3}\right). \tag{10}$$

The change in the position of *BV* from the reference line is equal to:

$$X_1 + BV_{c1} = X_1 + (X_3 - X_1)\left(\frac{V_1}{V_1 - V_3}\right). \tag{11}$$

2.1.2. Secondary condition-2

This condition is shown in Fig. 1 for two different values of $y - y_1$, where one value is positive and the other is negative. In either of these situations, *BV* cannot overtake *FV* before *OV* crosses *BV*. An accident can be avoided if *BV* does not change lane until *OV* crosses *BV*. The distance at which *BV* and *OV* meet is located somewhere between the initial positions of *BV* and *OV*. Thus, the change in the position of *BV* is given by:

$$BV_{c2} = R(x + y), \tag{12}$$

where R is a random number between 0 and 1.

The change in the position of *BV* from the reference line is equal to:

$$X_1 + BV_{c2} = X_1 + R(x + y) = X_1 + R(X_2 - X_1). \tag{13}$$

2.2. Primary condition-2

If *FV* is faster than *BV*, then it is not possible for *BV* to pass *FV*. Thus, the change in the position of *BV* is as follows.

$$X_1 + BV_{c3} = X_1 + Rx = X_1 + R(X_3 - X_1). \tag{14}$$

3. PVS to optimize a function

To implement the above theory of passing vehicles as an optimization process, a correlation needs to be obtained between the optimization terminology and the theory described above. Let us assume that the set of solutions comprise different vehicles on a two-lane highway. The value of the objective function or fitness value represents the velocity of vehicles, i.e., a solution with a better fitness value is considered to be a vehicle with a higher velocity. The design variables determine the position of the vehicle on a highway. Thus, PVS starts with a set of solutions referred to as a population of vehicles, from which three vehicles (solutions) are selected at random. Among these three selected vehicles, the current solution is correlated with *BV* and the other two can be designated randomly as *OV* and *FV*. The distances between the vehicles and their respective velocities are assigned based on the population size and its fitness value. After assigning the distances and velocities, the vehicles are checked to confirm the conditions for passing. Based on the conditions employed, the vehicles change their respective positions on the highway (change the current solution).

A step-wise procedure to implement PVS for the optimization of a given function is described in this section and PVS is explained with the aid of the flowchart in Fig. 2.

Step 1: Define the parameters

Population size (*PS*), stopping criteria (such as the number of generations (*NOG*), maximum function evaluations (FE), and error), number of design variables (*DV*), bounds on design variables (*LB, UB*)

Define the function for optimization

Minimize $f(X)$

Subject to $X = \{x_1, x_2, \ldots, x_{DV}\}$,

where $LB_i \le x_i \le UB_i$

Step 2: Generate the initial population

A population in PVS represents the number of vehicles on a two-lane highway and, mathematically, it represents the set of solutions.

$$X_{k\ldots PS,\ i\ldots DV} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,DV} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,DV} \\ \vdots & \vdots & & \vdots \\ x_{PS,1} & x_{PS,2} & \cdots & x_{PS,DV} \end{bmatrix}. \tag{15}$$

Step 3:

Select three vehicles (solutions), among which one solution represents the current vehicle (current solution) (*BV*) and the other two solutions are selected randomly.

Thus, select r_2 and r_3 randomly as follows.

$r_1 =$ current solution (*BV*), $X_{r1,i=1 \text{ to } DV}$

$r_2 \neq (r_1, 0) = R_1(PS)$, (OV), $X_{r2,\ i=1 \text{ to } DV}$

$r_3 \neq (r_1, r_2, 0) = R_2(PS)$, (FV), $X_{r3,\ i=1 \text{ to } DV}$

R_1, R_2 are random numbers $\in (0,1)$

Step 4:

Calculate the distances (*D*) and velocities (*V*) of the vehicles

The distances between the vehicles are calculated based on the fitness value. Arrange the population in ascending order and calculate the distance using Eq. (16). From the expressions, it can be observed that the value of the distance is normalized to 1. For example, if we consider a population size of 10, i.e., $PS = 10$, then the normalized distances for the 3rd, 6th, and 8th population members will be 0.3, 0.6, and 0.8 respectively, which ultimately indicate the distance from the datum. The velocity corresponding to the vehicle is calculated using Eq. (17). For the vehicles considered above, the values of the velocities are 0.7, 0.4, and 0.2, respectively. From these values, it can be seen that the best solution among the three is assigned a higher velocity (the 3rd solution is assigned 0.7 and the 8th solution is assigned 0.2).

$$D_k = \frac{1}{PS}(r_k) \tag{16}$$

$$V_k = R_k(1 - D_k), \tag{17}$$

where $k = (1,2,3)$and R is a random number $\in (0,1)$

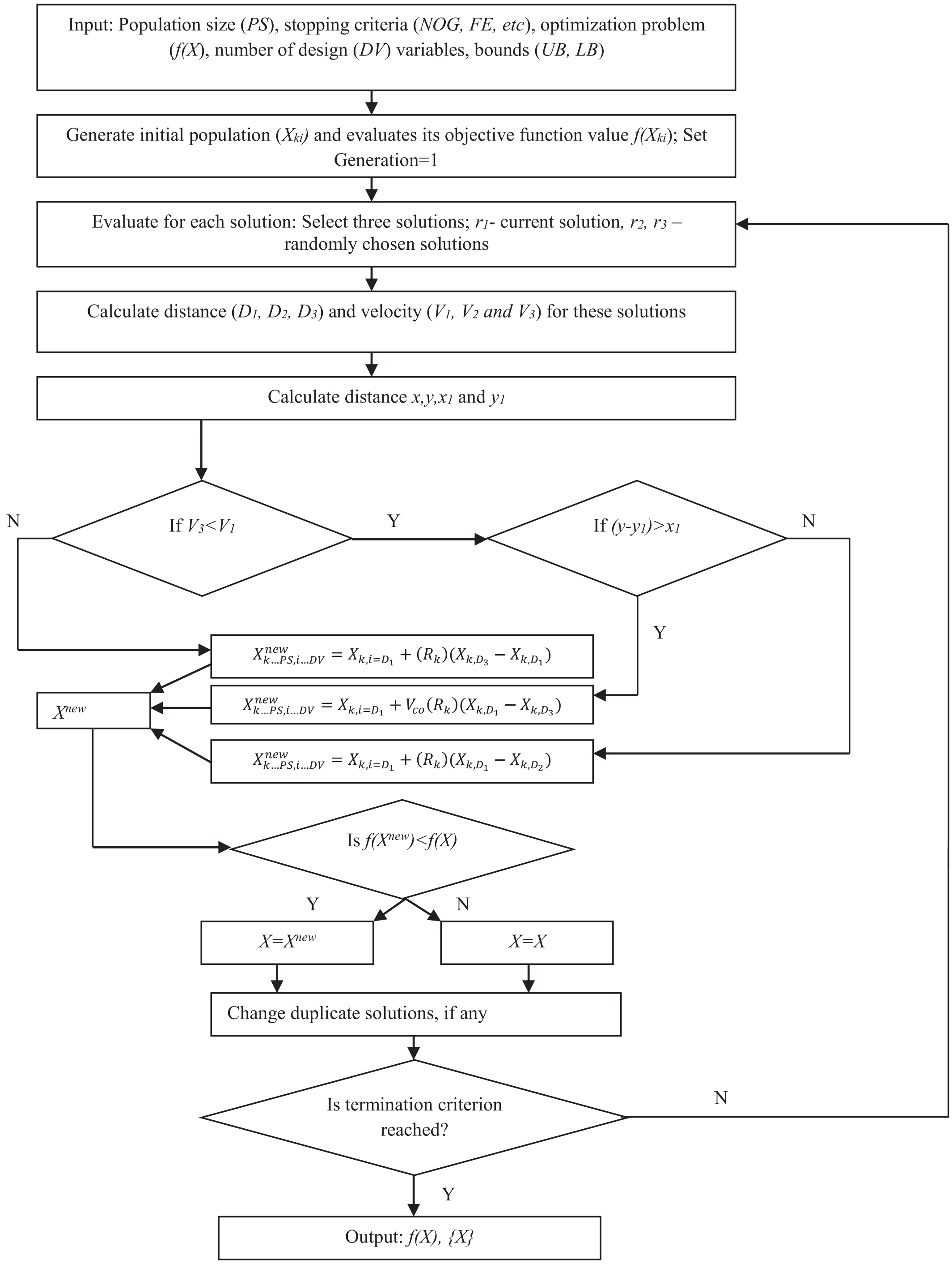

Fig. 2. Flowchart illustrating passing vehicle search (PVS).

Step 5:

Calculate the distance x, y, x_1 and y_1 using the following equations:

$$x = |D_3 - D_1| \tag{18}$$

$$y = |D_3 - D_2|, \tag{19}$$

and x_1, y_1 using Eqs. (4) and (7).

Step 6:

Update the solution
If $V_3 < V_1$
If $(y - y_1) > x_1$

$$X^{new}_{k\ldots PS,i\ldots DV} = X_{k,i=D_1} + V_{co}(R_k)\left(X_{k,D_1} - X_{k,D_3}\right). \tag{20}$$

Else

$$X^{new}_{k\ldots PS,i\ldots DV} = X_{k,i=D_1} + (R_k)\left(X_{k,D_1} - X_{k,D_2}\right). \tag{21}$$

End
Else

$$X^{new}_{k\ldots PS,i\ldots DV} = X_{k,i=D_1} + (R_k)\left(X_{k,D_3} - X_{k,D_1}\right). \tag{22}$$

End
where $R_i \in (0, 1)$,

$$V_{co} = \frac{V_1}{V_1 - V_3}. \tag{23}$$

Accept the solution if it is better than the previous solution.

Step 7:

Maintain diversity in the population by removing the duplicates as follows.

$$\begin{aligned} &\text{For } k = 1:2:PS \\ &\quad \text{If } X_k = X_{k+1} \\ &\quad i = rand^*(DV) \\ &\quad X_{k+1,i} = LB_i + rand^*(UB_i - LB_i). \\ &\quad \text{End if} \\ &\text{End for} \end{aligned} \tag{24}$$

Step 8:

Repeat the procedure until the termination criteria are satisfied.

4. Comparison of PVS with other meta-heuristics

Similar to other meta-heuristics (e.g., GA, PSO, Artificial Bee Colony (ABC), BBO, and TLBO), PVS is also a population-based method, which requires an initial set of solutions. All of these algorithms are based on specific natural principles, i.e., GA is based on the evolution of species, PSO is based on the behavior of swarms of birds, ABC is based on the foraging behavior of honey bees, BBO is based on the theory of immigration and the emigration of species, and TLBO is based on the teaching-learning philosophy. PVS uses the vector difference between the solutions in a similar manner to PSO, differential evolution (DE), ABC, TLBO, CSA, and FPA. It does not use information from the best solution to update the solutions, where this is similar to PSO, CSA, FPA, and TLBO. PVS uses greedy selection in a manner similar to ABC, CSA, and TLBO to accept the solutions. Unlike ABC and TLBO, PVS does not require different phases to update the solutions, such as the employed and onlooker bee phases in ABC, and teaching and learning phases in TLBO. Like DE, PVS uses three solutions to update the existing solutions, but the search expressions are different for these algorithms. A unique feature of this algorithm is that it uses three different mathematical expressions to update the solutions with different probabilities. Each different algorithm requires various algorithm parameters to operate correctly, i.e., GA requires the crossover probability and mutation probability, ABC requires the number of employed bees, onlooker bees, and limits, PSO requires learning factors, and CSA requires the probability and beta value. We only mention some algorithms because it is not possible to mention all algorithms in this study. These algorithm parameters influence the performance of the algorithm and appropriate values for these parameters must be tuned for particular applications. Sometimes, it is difficult to find appropriate values for these parameters and the user must often reach a compromise by selecting standard values from previous studies. By contrast, the TLBO algorithm does not require any algorithm parameters to function correctly. Like TLBO, PVS is also an algorithm parameter-free meta-heuristic optimization method, which is considered as one of the most effective features of this algorithm.

Table 1
Characteristics of different engineering design problems.

Sr. No.	Problems	No. of design variables	Continuous design variables	Discrete design variables	No. of constraints	Active constraints	F/S*	Objective
1	Pressure vessel	4	2	2	4	2	0.40	Minimize cost
2	Spring	3	3	0	4	2	0.01	Minimize weight
3	Welded beam	4	4	0	7	2	0.035	Minimize cost
4	Speed reducer	7	6	1	11	3	0.004	Minimize weight
5	Bearing	10	9	1	9	4	0.015	Maximize dynamic load carrying capacity
6	Multi-plate clutch	5	0	5	8	1	0.700	Minimize weight
7	Step cone pulley	5	5	0	11	4	0.000	Minimize weight
8	Robot gripper	7	7	0	7	1	0.025	Minimize difference in gripper force
9	Hydro-static bearing	4	4	0	7	3	0.003	Minimize power loss
10	Belleville spring	4	3	1	7	4	0.004	Minimize weight
11	Planetary gear train	9	0	9	11	0	0.000	Minimize error in gear ratio
12	Stiffened shell	5	0	5	5	0	0.300	Minimize cost
13	Four-stage gear box	22	0	22	86	26	0.000	Minimize weight

* F/S ratio between the feasible solutions in the search space (F) relative to the total search space(S).

5. Performance of PVS

In this section, we describe the performance of PVS with 13 dissimilar challenging engineering design optimization problems. We compared the performance of PVS with other well-known optimization techniques. PVS was coded in Matlab on an Intel® core™ i3 laptop with M350 @ 2.27 GHz. All of the constrained problems were converted into unconstrained problems using the static penalty method approach [57]. A penalty value was added to the objective function for each infeasible solution so it was penalized if the constraints were violated. This method is popular because it is simple to apply, where it only requires the addition of a penalty and it varies for different problems. An optimization problem is typically written as:

$$\begin{aligned}
&Minimize\ f(X),\\
&X = \{1, 2, \ldots, DV\}\\
&Subjected\ to:\\
&g_i(X) \le 0, \quad i = 1, \ldots, p\\
&h_i(X) = 0, \quad i = p+1, \ldots, NC,
\end{aligned}$$

where p is the total number of nonlinear constraints and NC is the total number of constraints. The constrained optimization problem can be converted into an unconstrained optimization problem using the static penalty approach as follows:

$$f(X) = f(X) + \sum_{i=1}^{p} P_i \max\{g_i(X), 0\} + \sum_{i=p+1}^{NC} P_i \max\{|h_i(X)| - \delta, 0\}, \tag{25}$$

where P_i is a penalty factor, which is generally assigned a large value.

6. Engineering design optimization problems

In this investigation, we considered 10 different widely used engineering design problems (pressure vessel, spring, welded beam, speed reducer, bearing, multi-plate clutch, step-come pulley, robot gripper, hydrostatic thrust bearing, and Belleville spring) and three additional challenging problems (4-stage gear box, stiffened welded shell, and planetary gear box). The characteristics of these problems are given in Table 1. The full details of these 10 widely used engineering problems have been described in several previous studies [30,36,58].

Matlab is a powerful tool for many engineering applications because it contains several ready-to-use toolboxes, among which “Optimization Toolbox” and “Genetic Algorithm and Direct Search (GADS) toolbox” can be used to solve optimization problems. It is interesting to note that comparisons of these toolboxes and other meta-heuristics have not been reported in previous studies. Thus, in this study, we compared the results obtained using the GA and SQP methods in the “GADS Toolbox” and “Optimization Toolbox,” respectively. SQP is a classical deterministic method, which gives the same solution (local optima) for a particular starting point. However, determining appropriate starting points to obtain the optimum solution (global solution) is quite difficult, and thus different starting points need to be investigated to obtain more confident solutions. In this study, we obtained results using 25 different starting points for SQP and 25 independent runs for the meta-heuristic algorithms.

The following parameters were considered for GA and SQP in Matlab,

For SQP: *DiffMinChange = 0.01, DiffMaxChange = 0.1, TolX = 1e–100, TolFun = 1e–100, TolCon = 1e–100, MaxFunEvals = 25,000, MaxIter= 25,000.*

Table 2
Results for the pressure vessel design optimization problem.

Algorithm		Best	Mean	Worst	Max_FE
Considering mixed variables					
PSO-DE	[65]	6059.714	6059.714	6059.714	42,100
TLBO	[36]	6059.714	6059.714	NA	20,000
ABC	[60]	6059.714	6245.308	NA	30,000
CVI-PSO	[67]	6059.714	6292.123	6820.41	25,000
BIANCA	[66]	6059.938	6182.002	6447.325	80,000
DEC-PSO	[61]	6059.714	6060.33	6090.526	300,000
BA	[62]	6059.714	6179.13	6318.95	20,000
CSA	[62]	6059.714	6447.73	6495.34	15,000
ISA	[64]	6059.714	6410.087	7332.84	5,000
FFA	[59]	6059.714	6064.33	6090.52	50,000
PVS		6059.714	6063.643	6090.526	42,100
		6059.714	6065.877	6090.526	20,000
		6059.714	6173.476	6410.09	15,000
		6062.244	6231.835	6824.409	5,000
GA (Matlab GA-toolbox)		6121.727	6854.328	7545.54	25,000
SQP (Matlab optimization toolbox)		NFS	NFS	NFS	25,000
Considering continuous variables					
NM-PSO	[68]	5930.314	5946.79	5960.056	80,000
WCA	[30]	5885.333	6198.617	6590.213	27,500
MBA	[69]	5889.321	6200.64	6392.5	70650
PVS		5885.333	5885.409	5886.035	20000

For GA: *PopulationSize* = *50, Generations* = *500, CrossoverFcn* = *@crossoverarithmetic,SelectionFcn* = *@selectionroulette, StallGenLimit* = *500,StallTimeLimit* = *120,TolFun* = *1e−100.*

We solve the pressure vessel design, spring design, welded beam design, speed reducer design, bearing design, and multi-plate clutch design problems using SQP, GA, and PVS. In addition, we solved the step-cone pulley design, robot gripper design, hydrostatic thrust bearing design, Belleville spring design, four-stage gear box design, stiffened welded shell design, and planetary gear design problems using SQP, GA, CSA, FPA, GSA, BBO, and PVS.

6.1. Pressure vessel design optimization problem

This problem requires the minimization of the total cost for a pressure vessel. The objective function is nonlinear and it is subject to three linear and one nonlinear inequality constraints. There are four design variables, where two are discrete and two are continuous. The first two discrete variables can obtain values that are multiples of 0.065. The ratio of feasible solutions relative to the search space (*F/S*) is approximately 0.40, which indicates the total feasible region in the search space. The ratio *F/S* is obtained using 100,000 random solutions in the search space. The best known solution for this problem is $f(X) = 6059.714335$ at $X = \{0.8125, 0.4375, 42.0984456, 176.6365958\}$. At the optimum solution, there are two active constraints. This problem has been addressed by many researchers using different optimization techniques and their variants. Some of the basic methods and variants that have been applied to this problem include PSO-DE, TLBO, ABC, constraint violation with interval arithmetic PSO (CVI-PSO), multi-population GA (BIANCA), diversity-enhanced constrained PSO (DEC-PSO), BA, CSA, interior search algorithm (ISA), firefly algorithm (FFA), WCA, Nelder–Mead PSO (NM_PSO), and mine blast algorithm (MBA) [59–61,30,62–67,36,68,69]. Some researchers have solved this problem using only continuous design variables [68,30,69]. Thus, we obtained the results using PVS for two cases: (a) with discrete design variables and (b) with continuous design variables. A population size of 50 was considered for PVS and the stopping criterion was 42,100 FE. Various studies have used different number of FEs to investigate the performance of their proposed methods. Thus, we compared the results obtained using PVS with 5000, 15,000, 20,000, and 42,100 FEs to the results produced by other algorithms. The results were mainly compared based on the mean solutions obtained in 25 different independent runs. The results are summarized in Table 2, which show that PVS is capable of finding the global solution for the pressure vessel problem. The mean of the solutions obtained in 25 runs was better at 5000 FEs than that by ISA, better at 15,000 FEs than that by CSA, and better at 20,000 FEs than that by BA. However, PVS obtained inferior results to TLBO and PSO-DE at 20,000 and 42,100 FEs. Moreover, PVS performed better than all of the other algorithms. PVS outperformed WCA, MBA, and NM-PSO in terms of the mean and worst solutions when using continuous design variables. Moreover, SQP failed to find a feasible solution with 25 different starting points. The results obtained using GADS toolbox were also not adequate because it failed to find the global solution and its mean performance was also poor compared with the other methods.

6.2. Spring design optimization problem

The problem requires the minimization of the weight of a tension/compression spring by considering constraints on the minimum deflection, shear stress, and surge frequency, as well as limits on the geometric dimensions. There are three continuous design variables: wire diameter (x_1), mean coil diameter (x_2), and number of active coils (x_3). The optimum solution reported in

Table 3
Results for the spring design optimization problem.

Algorithm		Best	Mean	Worst	Max_FE
PSO–DE	(Liu et al.)	0.012665	0.012665	0.012665	42,100
TLBO	[36]	0.012665	0.012666	NA	20,000
ABC	[60]	0.012665	0.012709	NA	30,000
WCA	[30]	0.012665	0.012746	0.012952	11,750
		0.012665	0.013013	0.015021	2000
CVI-PSO	[67]	0.012666	0.012731	0.012843	25,000
BIANCA	[66]	0.012671	0.012681	0.012913	80,000
BA	[62]	0.012665	0.013501	0.016895	20,000
MBA	[69]	0.012665	0.012713	0.012900	7,650
ISA	[64]	0.012665	0.012799	0.013165	8000
FFA	[59]	0.012665	0.012677	0.000013	50,000
PVS		0.012665	0.012665	0.012665	42,100
		0.012665	0.012666	0.012667	20,000
		0.012665	0.012670	0.012710	8,000
		0.012680	0.012838	0.013141	2,000
GA (Matlab GA-toolbox)		0.012671	0.012683	0.012693	25,000
SQP (Matlab Optimization toolbox)		0.012928	0.017951	0.032122	25,000

Table 4
Results for the welded beam design optimization problem.

Algorithm		Best	Mean	Worst	Max_FE
PSO–DE	[65]	1.724852	1.724852	1.724852	66,600
TLBO	[36]	1.724852	1.728447	NA	20,000
ABC	[60]	1.724852	1.741913	NA	30,000
WCA	[30]	1.724856	1.726427	1.744697	46,450
		1.724857	1.73594	1.801127	30,000
CVI-PSO	[67]	1.724852	1.725124	1.727665	25,000
BIANCA	[66]	1.724852	1.752201	1.793233	80,000
BA	[62]	1.7312	1.878656	2.345579	20,000
MBA	[69]	1.724853	1.724853	1.724853	47,340
FFA	[59]	1.724852	1.724852	1.724852	50,000
PVS		1.724852	1.724852	1.724852	50000
		1.724852	1.724887	1.725056	20,000
GA (Matlab GA-toolbox)		2.026769	2.76033	3.162137	25,000
SQP (Matlab optimization toolbox)		2.124207	4.096846	5.493251	25,000

previous studies is $f(X) = 0.012665$ with $X = \{0.051749, 0.358179, 11.203763\}$. This optimum solution has two active constraints. The total feasible region accounts for nearly 1% of the search space. This problem has been solved by many researchers using different optimization techniques, including PSO-DE, TLBO, ABC, WCA, CVI-PSO, BIANCA, BA, MBA, ISA, and FFA. Different numbers of FEs were also used to investigate the performance of their proposed methods. Thus, we investigated PVS using 2000, 8000, 20,000, and 42,100 FEs, with a population size of 50. The results obtained in 25 independent runs are summarized in Table 3, which show that PVS can find the optimum solution to the spring design problem. For this problem, GA and SQP performed poorly compared with other meta-heuristics when obtaining the optimum and mean solutions. The minimum number of FEs required using WCA was 2000 according to Eskandar [30]. Our results show that PVS failed to find the optimum solutions in 2000 FEs over 25 runs, but PVS outperformed WCA in terms of the mean result. With 8000 FEs, PVS performed better than ISA and MBA, while with 20,000 FEs, PVS performed better than BA and it was equivalent to TLBO. In addition, with 42,100 FEs, the performance of PVS was almost the same as that of PSO-DE. PVS performed better than the other algorithms in terms of the solution quality and computational effort.

6.3. *Welded beam design optimization problem*

This problem requires the optimization of the cost of a welded beam by considering four design variables: height of weld (x_1), length of weld (x_2), height of beam (x_3), and width of beam (x_4). Constraints are considered for the shear stress, bending stress, buckling load on the bar, end deflection, and side constraints. This problem comprises nearly 3.5% of the feasible region in the search space and the optimum value reported previously is $f(X) = 1.724852$ at $X = \{0.205730, 3.470489, 9.036624, 0.205730\}$ with two active constraints. This problem has been solved using different optimization techniques such as PSO–DE, TLBO, ABC, WCA, CVI-PSO, BIANCA, BAT, MBA, and FFA. We solved this problem by considering a population size of 50 with 20,000 and 50,000 FEs. The results are summarized in Table 4, which show that at 20,000 FEs, PVS performed better than BA and TLBO, while at 50,000 FEs, the performance of PVS was almost the same as that of FFA and MBA. The average performance (based on the mean value)

Table 5
Results for the speed-reducer design optimization problem.

Algorithm		Best	Mean	Worst	Max_FE
Design variable bounds-1					
PSO–DE	[65]	2996.348167	2996.348174	2996.348204	54,350
ABC	[60]	2997.058	2997.058	NA	30,000
TLBO	[36]	2996.34817	2996.34817	NA	20,000
CSA	[62]	3000.98	3007.1997	30090	5,000
FFA	[59]	2996.37	2996.51	2996.669	50,000
PVS		2996.4	2996.48271	2996.7123	5,000
		2996.348165	2996.350001	2996.366218	20,000
		2996.348165	2996.348165	2996.348165	54,350
GA (Matlab GA-toolbox)		2996.415435	2996.627632	2997.57529	25,000
SQP (Matlab Optimization toolbox)		3008.465029	3083.238067	3160.475465	25,000
Design variable bounds-2					
DELC	[75]	2994.471066	2994.471066	2994.471066	30,000
WCA	[30]	2994.471066	2994.474392	2994.505578	15,150
MBA	[69]	2994.4824	2996.769	2999.65	6,300
PVS		2994.47326	2994.7253	2994.8327	6,000
		2994.471066	2994.483258	2994.752395	15,150
		2994.471066	2994.472059	2994.477593	30,000

by PVS was better than that of the other algorithms. In addition, SQP and GADS toolbox also obtained inferior performance in this problem.

6.4. Speed reducer problem

The objective of this problem is to minimize the weight subject to different constraints on the bending stress, surface stress, transverse deflections of the shafts, and stresses in the shafts. The seven design variables considered are: face width (x_1), module of teeth (x_2), number of teeth in the pinion (x_3), length of the first shaft between bearings (x_4), length of the second shaft between bearings (x_5), and the diameters of the first and second shafts (x_6,x_7). The third variable (number of teeth) is an integer, whereas the others are continuous. This problem comprises nearly 0.4% of the feasible region and the optimum solution reported previously is $f(X) = 2996.3481$ at $X = \{3.49999, 0.6999, 17, 7.3, 7.8, 3.3502, 5.2866\}$ with three active constraints. This problem has been solved by considering two different ranges for the design variable "x_5" in previous studies: (a) $7.8 \le x_5 \le 8.3$ (design variable range-1) and (b) $7.3 \le x_5 \le 8.3$ (design variable range-2). We solved the problem under design variable-1 using PVS with 5000, 20,000, and 54,350 FEs, and with a population size of 50. The results are shown in Table 5, which demonstrate that the performance of PVS was better than CSA with 5000 FEs and almost the same as TLBO with 20,000 FEs. PVS performed better than the other algorithms in terms of the solution quality and computational effort. We solved the problem under design variable-2 using 6000, 15,150, and 30,000 FEs with a population size of 50. The results show that at 6000 FEs, PVS performed better than MBA, and at 15,150 FEs, the performance of PVS was almost the same as WCA. At 30,000 FEs, the performance of PVS was almost the same as differential evolution level comparison (DELC). In addition, GADS and Optimization Toolbox failed to obtain optimum solutions for this problem.

6.5. Bearing design optimization problem

The objective of this problem is to maximize the dynamic load carrying capacity by considering 10 geometric design variables and nine constraints based on assembly and geometric limitations. Among the 10 design variables, one design variable (the number of balls in the bearing) is required to attain an integer value. This problem comprises nearly 1.5% of the feasible region and the best known solution for this problem is $f(X) = 81{,}859.74$ at $X = \{21.42559, 125.7191, 11, 0.515, 0.515, 0.424266, 0.633948, 0.3, 0.068858, 0.799498\}$, which has four active constraints. This problem was solved using various optimization techniques such as GA, ABC, TLBO, WCA, MBA, and multi-member diversity-based DE (MDDE). For PVS, we used 10,000 and 20,000 FEs with a population size of 50. The results are summarized in Table 6, which show that PVS performed better than GA and ABC. In addition, PVS performed slightly better than TLBO, but worse than MDDE in terms of the mean solution. WCA and MBA obtained better results than all of the other algorithms, but the results were not comparable because the number of balls (x_3) has to be an integer value, whereas it is considered to be continuous by WCA and MBA.

6.6. Multi-plate disc clutch brake optimization problem

This problem requires the minimization of the weight of a multiple disc clutch brake by considering five discrete design variables: inner radius, outer radius, thickness of discs, actuating force, and number of friction surfaces. There are eight different constraints based on the geometry and operating conditions. The feasible region comprises nearly 70% of the search space. The optimum solution for this problem is $f(X) = 0.313656611$ at $X = \{70, 90, 1, 810, 3\}$ with one active constraint. We compared

Table 6
Results for the bearing design optimization problem.

Algorithm		Best	Mean	Worst	Max_FE
GA4	[76]	81843.3	NA	NA	225,000
ABC	[36]	81859.74	81496	78897.81	10,000
TLBO	[36]	81859.74	81438.98	80807.85	20,000
WCA	[30]	85538.48	83847.16	83942.71	3950
MBA	[69]	85535.9611	85321.4030	84440.1948	15100
MDDE	[77]	81858.83	81848.7	81701.18	10,000
PVS		81859.59	80803.57	78897.81	10,000
		81859.74	81550	79834.79	20,000
GA (Matlab GA-toolbox)		81822.53	80588.51	79279.29	25000
SQP (Matlab optimization toolbox)		74358.04	12% FS	12% FS	25000

Table 7
Results for the multi-plate disc clutch brake optimization problem.

Algorithm		Best	Mean	Worst	Max_FE
ABC	[36]	0.313657	0.324751	0.352864	600
TLBO	[36]	0.313657	0.327166	0.392071	1200
WCA	[30]	0.313657	0.313657	0.313657	500
PVS		0.313657	0.333652	0.352864	600
		0.313657	0.328163	0.392071	1200
GA (Matlab GADS-toolbox)		0.313657	0.330712	0.401873	25000
SQP (Matlab optimization toolbox)		1.08682	1.939037	3.133503	25000

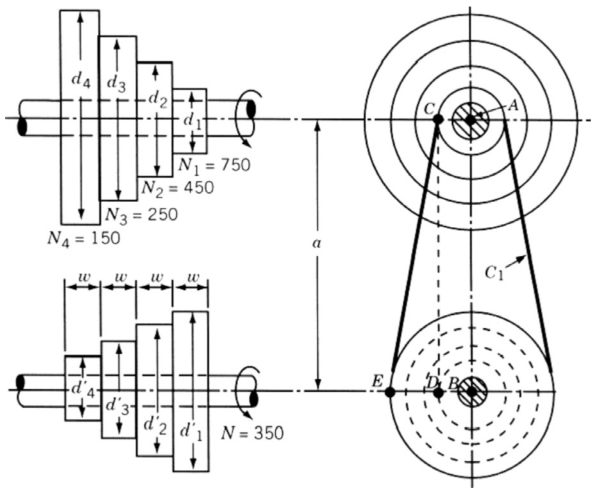

Fig. 3. Step cone pulley [36].

the results obtained by PVS and other optimization techniques such as ABC, TLBO and WCA, which are shown in Table 7. PVS was implemented using a population size of 20 with 600 and 1200 FEs. The results showed that the performance of PVS was almost the same as that of TLBO but the results were inferior to those obtained by WCA and ABC in terms of the mean solutions. However, PVS could find optimum solutions at 600 FEs in 25 independent runs.

6.7. Step-cone pulley

A step-cone pulley needs to be designed with the minimum weight according to five design variables, where four design variables are diameters (d_1, d_2, d_3, and d_4) in each step and the last is the width of the pulley (w), as shown in Fig. 3. The design is subject to 11 constraints, where three are equality constraints and the others are inequality constraints, which are used to ensure the same belt length in all of the steps, tension ratios, and power transmitted by the belt. The optimum solution reported previously is $f(X) = 16.63450513$ at $X = \{40, 54.76430219, 73.01317731, 88.42841977, 85.98624273\}$ with four active constraints. This problem is considered to be one of the most challenging optimization problems because the ratio of the feasible region relative to that of the total search space is almost 0.0001 and it also includes three equality constraints, which increase the difficulty of the problem. This problem was solved using PVS, CSA [11], FPA [25], GSA [28], and BBO [7] by considering 15,000 FEs

Table 8
Results for the step-cone pulley, robot gripper, hydrostatic thrust bearing, and Belleville spring design optimization problems (best solutions obtained in 25 runs).

	Step-cone pulley	Robot gripper	Hydrostatic bearing	Belleville spring
Max_FE	15000	25000	25000	15000
TLBO [36]	16.63451	4.247644	1625.443	1.979675
ABC [36]	16.63466	4.247644	1625.443	1.979675
MDDE [77]	14.488	NA	1638.40	1.979675
BBO (2008)	48.95349	4.774746	2573.441	2.081124
CSA (2009)	16.72545	4.247644	1713.029	1.981267
GSA (2009)	119.2703	4.598523	3093.021	2.400689
FPA (2012)	16.7416	4.24764	1837.852	1.984885
PVS	16.63496	4.247644	1625.443	1.979688
GA (Matlab GADS-toolbox)	21.79609	4.833581	2414.68	2.046987
SQP (Matlab Optimization toolbox)	58.3861	NFS	NFS	2.418122

Table 9
Results for the step-cone pulley, robot gripper, hydrostatic thrust bearing, and Belleville spring design optimization problems (mean solutions obtained in 25 runs).

	Step-cone pulley	Robot gripper	Hydrostatic bearing	Belleville spring
Max_FE	15000	25000	25000	15000
TLBO [36]	24.01136	4.937701	1797.708	1.979687
ABC [36]	36.0995	5.086611	1861.554	1.995475
MDDE [77]	16.7256	NA	1759.10	1.979675
BBO (2008)	70.55566	5.257477	3605.523	2.229694
CSA (2009)	17.15562	4.465418	1900.772	1.988929
GSA (2009)	192.7157	7.146706	3913.656	2.666064
FPA (2012)	17.1318	4.71694	1930.239	1.990434
PVS	20.03032	4.636115	1832.492	1.983524
GA (Matlab GADS-toolbox)	60.07632	5.676179	3339.823	2.19501
SQP (Matlab optimization toolbox)	4666.637	NFS	NFS	8% FS

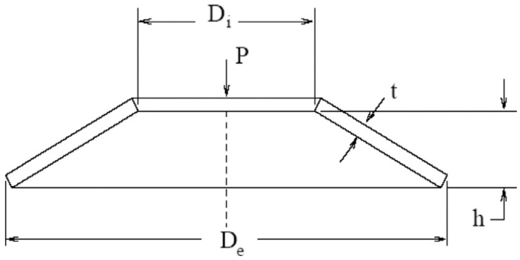

Fig. 4. Belleville spring [73].

with a population size of 50, and using GA and SQP (from Matlab toolbox) by considering 25,000 FEs. Our results are compared with the published results obtained by TLBO, ABC, and MDDE in Tables 8 and 9.

6.8. *Belleville spring*

This problem requires the minimization of the weight of a Belleville spring by considering four design variables: thickness of the spring, height of the spring, external diameter of the spring, and internal diameter of the spring, as shown in Fig. 4. Among these design variables, the thickness of the spring is a discrete variable whereas the others are continuous variables. The design is subject to constraints on the compressive stress, deflection, height to deflection ratio, height to maximum height ratio, outer diameter, inner diameter, and slope of the spring. The feasible region comprises nearly 0.4% of the total search space and the best known solution is $f(X) = 1.979674758$ at $X = \{0.204143354, 0.2, 10.03047329, 12.01\}$ with four active constraints. We solved this problem using PVS, CSA, FPA, GSA, and BBO with a population size of 50 and 15,000 FEs, and the results are shown in Tables 8 and 9.

6.9. *Robot gripper*

The robot gripper problem requires the minimization of the difference between the maximum and minimum force applied by a gripper for a range of gripper end displacements by considering seven continuous design variables and the geometric dimensions, as shown in Fig. 5. The design is subject to six different geometric constraints, where the feasible region comprises

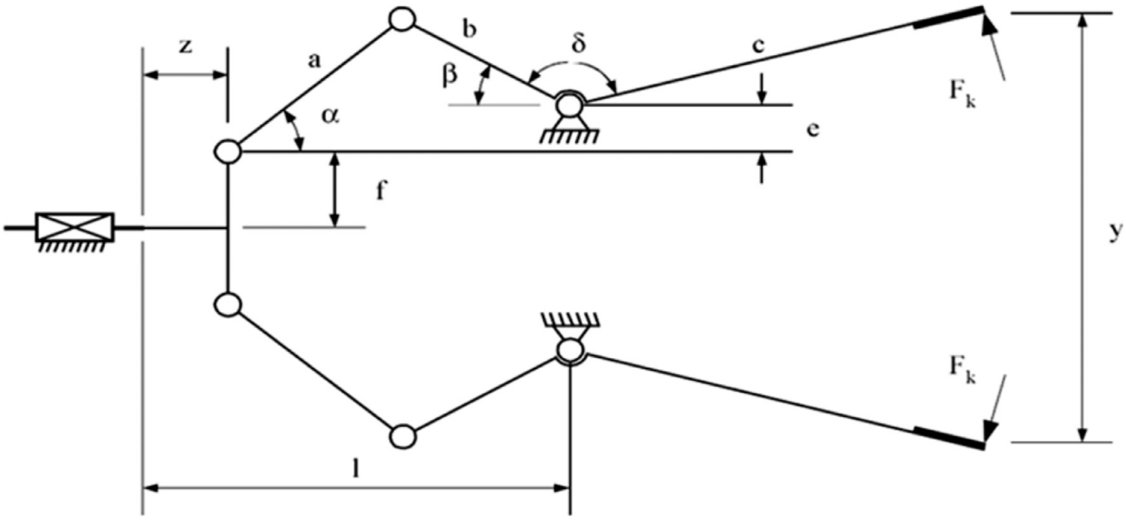

Fig. 5. Robot gripper [36].

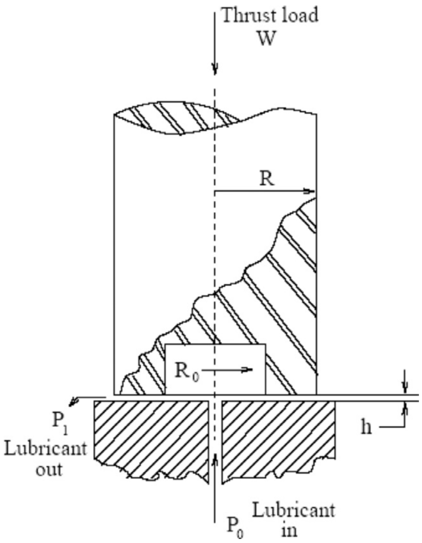

Fig. 6. Hydrostatic thrust bearing [36].

almost 2.5% of the total search space. The optimum solution reported previously is $f(X) = 4.247643634$ at $X = \{150, 150, 200, 0, 150, 100, 2.339539113\}$ with one active constraint. Moreover, the optimum solution occurs at the bound points of the design variables, except for the last design variable, which is one of the unique features of this problem. We solved this problem using 25,000 FEs and a population size of 50 with PVS, CSA, FPA, GSA, and BBO. The results are summarized in Tables 8 and 9.

6.10. Hydrostatic thrust bearing

In this problem, the hydrostatic thrust bearing needs to be designed to achieve the minimum power loss with four design variables: bearing step radius (R), recess radius (Ro), oil viscosity (μ), and flow rate (Q), as shown in Fig. 6. The design is subject to seven different constraints, i.e., the load carrying capacity, inlet oil pressure, oil temperature rise, oil film thickness, and physical constraints, where the feasible region comprises almost 0.3% of the total search space. The optimum solution reported previously is $f(X) = 1625.4427649821$ at $X = \{5.95578050261541, 5.38901305194167, 0.00000535869726706299, 2.26965597280973\}$. This problem is also a challenging because it is highly sensitive to the design variables. This problem was solved by PVS, CSA, FPA, GSA, and BBO using 25,000 FEs with a population size of 50.

Meta-heuristic methods require specific algorithm parameters to operate effectively. However, these parameters hinder the quality of the solution and the search capacity of the algorithm. These parameters are problem-specific and they might only ensure that a particular set of parameters works effectively with a specific class of problems, whereas they could fail with others.

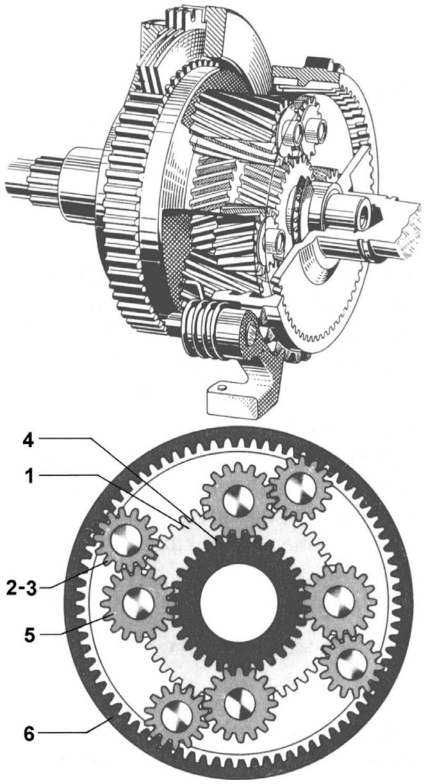

Fig. 7. Planetary gear train. 1 – small sun gear; 2 and 3 – broad planet gear; 4 – large sun gear; 5 – narrow planet gear; 6 – ring gear [74].

Thus, these parameters are determined after conducting experiments using different values, and thus the set of parameters that obtains better performance is considered for further investigations. In this study, the parameters considered for the different algorithms were as follows: for BBO, maximum immigration rate (I_{max}) = 1, minimum immigration rate (I_{min}) = 0, and mutation coefficient (m) = 0.05; for GSA, gravitation constant (G_0) = 100, gravitation reduction factor (α) = 20, and normalizing factor (F_{norm}) = 2; for CSA, probability factor (pa) = 0.25 and Levy flight factor (β) = 1.5; and for FPA, probability switch factor (p_a) = 0.25 and Levy flight factor (β)=1.5.

According to the results shown in Tables 8 and 9, for the step-cone pulley design problem, the performance of PVS was almost the same as that of TLBO and ABC in obtaining the optimum solution. PVS performed better than TLBO and ABC in terms of the mean function obtained in 25 runs. CSA and FPA outperformed all of the algorithms in terms of the mean function value, but they were unable to find the optimum solution. For the robot gripper problem, excluding BBO and GSA, all of the algorithms performed equally well at finding the global solution, but CSA obtained better results than all of the other algorithms at finding the mean function value, although PVS was ranked second best. For the hydrostatic thrust bearing problem, PVS could find the optimum solution in 25 runs, whereas CSA and FPA failed to obtain the optimum solution. TLBO outperformed all of the algorithms at finding the mean solutions, but it required twice the number of FEs to compute the values. For the Belleville spring problem, PVS, ABC, and TLBO achieved almost the same performance. The results also showed that GA and SQP performed poorly in all of the problems. Indeed, SQP failed to find feasible solutions to the robot gripper and hydrostatic bearing problems. Furthermore, SQP reported only 8% of the feasible solutions for the Belleville spring design problem.

We also evaluated the performance of PVS by considering three more challenging engineering design problems, i.e., a planetary gear train, stiffened welded shell, and four-stage gear box, where all of the design variables are discrete and the problems have many local optimum solutions in a very restricted feasible region. Detailed mathematical expressions for these problems are given in the Appendix.

6.11. Planetary gear train design optimization problem

This problem was formulated by Simionescu et al. [70] and the objective is to synthesize the gear-teeth number for an automatic planetary transmission system, as shown in Fig. 7, which is used in automobiles to minimize the maximum errors in the gear ratio. This problem considers six design variables based on the number of teeth in the gears (N_1, N_2, N_3, N_4, N_5, and N_6), which can only take integer values. There are three more discrete design variables: number of planet gears (P) and modules of the gears (m_1 and m_2), which can only take specified discrete values. The problem is subject to 11 constraints for different assembly and geometric restrictions, one of which is an equality constraint. The ratio of feasible region relative to the total search space

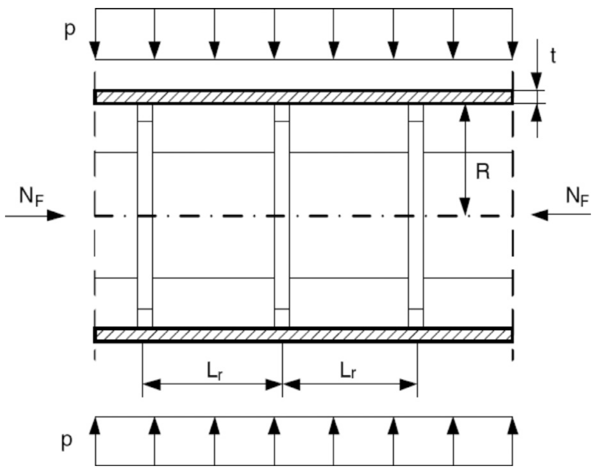

Fig. 8. Stiffened welded cylindrical shell with stringer and ring stiffener acted upon by compression and external pressure [71].

Table 10
Results for the four-stage gear box, welded stiffened shell, and planetary gear train (best solutions obtained in 25 runs).

Best	Four-stage gear box	Stiffened welded shell	Planetary gear box
Max_FE	25000	25000	25000
ABC-GA [73]	55.494494	55724.831	0.53
ABC-DE [73]	59.763563	55326.293	0.527814
ABC-BBO [73]	46.623205	55326.293	0.52735
TLBO [73]	43.792433	55326.293	0.52735
ABC [73]	49.836165	55326.29341	0.525769
BBO (2008)	39.286301	55724.831	0.532222
CSA (2009)	37.289069	55326.293	0.526281
GSA (2009)	44.431698	55944.771	0.53623
FPA (2012)	37.40714	55326.293	0.52325
PVS	37.267472	55326.293	0.525588
GA (Matlab GADS-toolbox)	NFS	57165.985	0.569018
SQP (Matlab Optimization toolbox)	NFS	NFS	NFS

is less than 0.0001 and the best solution previously is $f(X) = 0.525$ with $X = \{40, 21, 14, 19, 16, 69, 5, 2.25, 2.5\}$, with one active constraint.

6.12. Stiffened welded shell design optimization problem

This problem was formulated by Jarmai et al. [71] and it requires the optimization of the cost for a cylindrical shell member, which is orthogonally stiffened using ring stiffeners and stringers with a halved I-section, as shown in Fig. 8. There are five design variables: shell thickness (t), number of longitudinal stiffeners (stringers) (n_s), number of ring stiffeners (n_r), box height (h_r), and stringer stiffness height (h), which can only have specified discrete values. The design is subject to five different constraints to consider buckling and manufacturing restrictions. The feasible region comprises nearly 30% of the total search space and the best value was reported by Jarmai et al. [71] as $f(X) = 55326.29$ with $X = \{14, 27, 10, 250, 203\}$ and no active constraints.

6.13. Four-stage gear box

This problem was introduced by Pomrehn and Papalambros [72], and the objective is to minimize the weight of a gear box. There are 22 design variables, including the number of teeth, the blank thickness, and the positions of the gear and pinion. All of the design variables are discrete and eight are integer variables. This problem is subject to 86 different constraints on the strength of the gears, contact ratio, size of gears, assembly of gears, pitch, and kinematics, where the ratio of the feasible region relative to the total search space is less than 0.0001. This problem is challenging because there are many local solutions. The best solution reported previously is $f(X) = 36.57$ at $X = \{19, 19, 20, 22, 38, 49, 42, 40, 3.175(1, 1, 1, 1,) 12.7(7, 4, 6, 3, 5, 7, 7, 4, 5, 7)\}$, with 25 active constraints.

We investigated these three problems using PVS, CSA, FPA, GSA, and BBO by considering 25,000 FEs with a population size of 50. The results are illustrated in Tables 10 and 11, where they are also compared with the results obtained in previous studies using TLBO, ABC, Hybrid DE, BBO, and GA. For the planetary gear design problem, PVS and ABC achieved almost equal performance and they were better than all of the other algorithms. The performance of PVS was inferior to those of ABC, FPA, and ABC-BBO

Table 11
Results for the four-stage gear box, welded stiffened shell, and planetary gear train (mean solutions obtained in 25 runs).

	Four-stage gear box	Stiffened welded shell	Planetary gear box
Max_FE	25000	25000	25000
ABC-GA [73]	8% FS	56920.4	0.53669
ABC-DE [73]	24% FS	55852.8	0.54479
ABC-BBO [73]	63% FS	55921.9	0.52908
TLBO [73]	53% FS	55667.4	0.53371
ABC [73]	14% FS	55883.37	0.527292
BBO	76% FS	56783.8	0.57247
CSA	31 % FS	55559.8	0.53167
GSA	19% FS	56746	0.73245
FPA	63% FS	55590.9	0.52978
PVS	88% FS	55531.2	0.53063
GA (Matlab GADS-toolbox)	NFS	60271.6	32% FS
SQP (Matlab Optimization toolbox)	NFS	NFS	NFS

Table 12
Friedman rank test results for the engineering design problems.

	PVS	GA	TLBO	ABC	CSA	FPA	GSA	BBO
For best solutions								
Friedman rank value	18	50	22	20.5	24.5	23	50	44
Normalized rank value	1.00	2.78	1.22	1.14	1.36	1.28	2.78	2.44
Rank	1	7	3	2	5	4	7	6
For mean solutions								
Friedman rank value	14	49	23	31	21	21	51	42
Normalized rank value	1	3.50	1.64	2.21	1.50	1.50	3.64	3.00
Rank	1	6	3	4	2	2	7	5

in terms of the mean solutions, but PVS obtained better results than the remaining algorithms. For the stiffened welded shell design problem, all of the algorithms performed poorly at finding the optimum solution except for GSA and BBO. The average performance was almost the same for PVS, CSA, and FPA, but they were better than the other algorithms. For the four-stage gear box design problem, all of the algorithms struggled to find feasible solutions. All of the algorithms actually failed to obtain feasible solutions to this particular problem, so it was not fair to compare the mean solutions. Thus, we compared the ability of the algorithms to find feasible solutions. According to the results of this comparison, PVS performed better at finding feasible solutions compared with the other algorithms. PVS obtained the best solution compared with the other algorithms, although not the optimum. The results obtained by CSA and FPA were almost the same as those with PVS. GA and SQP performed poorly in all three of these challenging engineering design problems.

According to the results, it was very difficult to identify the best performing algorithm because the different algorithms exhibited variable performance in each problem. Some algorithms performed better in some cases but failed in others. Thus, to quantify the performance of the algorithms and to rank them, we applied the Friedman rank test [41]. This test was performed separately using the best solutions and the mean solutions for CSA, FPA, ABC, TLBO, GSA, BBO, and PVS. The results of the Friedman rank test are summarized in Table 12. The results show that the Friedman rank value was lowest for PVS followed by ABC and TLBO in obtaining the best solution (optimum solution). There were marginal differences between the values for TLBO, ABC, CSA, and FPA. The rank was obtained based on the Friedman rank value, which gave PVS the highest rank. The Friedman test based on the mean solutions indicated that PVS was ranked first followed by CSA and FPA in joint second position, with TLBO in the fourth position. The algorithms were ranked but it was still difficult to identify significant differences in performance among these algorithms. Thus, the Holm-Sidak multiple comparisons test was employed to test for significant differences among the algorithms [42]. We tested PVS, TLBO, ABC, CSA, and FPA because these algorithms obtained better performance in the problems considered. We did not consider GSA, BBO, and GA in this test due to their poor performance in the problems considered. The results of the Holm-Sidak test are given in Table 13, which show the *p*-values of multiple pairwise comparisons. A higher *p*-value indicates no significant difference in the results for two algorithms. The results show that TLBO, CSA, and FPA yielded similar performance to PVS, but there was a significant difference in the results obtained by PVS and ABC. Similar to the previous results in Tables 8–11, TLBO, CSA, and FPA achieved almost the same performance and their higher *p*-values indicated that there was no significant difference among these algorithms. The best results and the design variables obtained using PVS are listed in Tables 14 and 15.

It is also necessary to determine the performance of proposed optimization methods in terms of the convergence of the algorithm and the computational efforts in terms of the time consumed by running the algorithm. The convergence of all the algorithms varied according to the problem and thus it would have been unsuitable to present the convergence for all of the problems considered. Thus, we employed a combined convergence strategy by considering different problems in this analysis.

Table 13
Holm-Sidak test for engineering design problems.

Algorithms*	*p*-value
1–3	0.03678
2–3	0.13381
3–4	0.14562
3–5	0.18356
1–5	0.40813
1–4	0.48621
1–2	0.51581
2–5	0.85681
4–5	0.89399
2–4	0.96235

* 1 = PVS, 2 = TLBO, 3 = ABC, 4 = CSA, 5 = FPA

Table 14
Best results obtained by PVS for the pressure vessel, spring, speed reducer, bearing, and multi-plate disc clutch brake problems.

	Pressure vessel		Spring	Welded beam	Speed reducer		Bearing	Multi-plate disc clutch brake
	Discrete	Continuous			Range-1	Range-2		
$f(X)$	**6059.71434**	**5885.33277**	**0.01267**	**1.72485**	**2996.34817**	**2994.47107**	**81859.74121**	**0.31366**
x_1	0.81250	0.77817	0.05169	0.20573	3.50000	3.50000	21.42559	70.00000
x_2	0.43750	0.38465	0.35680	3.47049	0.70000	0.70000	125.71906	90.00000
x_3	42.09845	40.31962	11.28442	9.03662	17.00000	17.00000	11.00000	1.00000
x_4	176.63660	200.00000	0.01267	0.20573	7.30000	7.30000	0.51500	980.00000
x_5			0.05171		7.80000	7.71532	0.51500	3.00000
x_6			0.35721		3.35021	3.35021	0.40043	
x_7			11.26005		5.28668	5.28665	0.68016	
x_8							0.30000	
x_9							0.07999	
x_{10}							0.70000	

Table 15
Best results obtained by PVS for the step-cone pulley, Belleville spring, robot gripper, hydrostatic thrust bearing, planetary gear train, welded stiffened shell, and four-stage gear box problems.

	Step-cone pulley	Belleville spring	Robot gripper	Hydrostatic thrust bearing	Planetary gear	Welded stiffened shell	Four-stage gear box
$f(X)$	**16.63451**	**1.97967**	**4.24764**	**1625.44386**	**0.52559**	**55326.29341**	**37.26747**
x_1	40.00000	0.20414	150.00000	5.95578	34	14	23
x_2	54.76430	0.20000	150.00000	5.38901	25	27	23
x_3	73.01318	10.03047	200.00000	0.00001	33	10	21
x_4	88.42842	12.01000	0.00000	2.26966	32	250	21
x_5	85.98624		150.00000		23	203	53
x_6			100.00000		116		49
x_7			2.31187		4		42
x_8					2.5		42
x_9					1.75		3.175
x_{10}							3.175
x_{11}							3.175
x_{12}							3.175
x_{13}							76.2
x_{14}							38.1
x_{15}							76.2
x_{16}							76.2
x_{17}							76.2
x_{18}							88.9
x_{19}							63.5
x_{20}							50.8
x_{21}							88.9
x_{22}							50.8

First, we analyzed the results obtained in different independent runs by considering the same initial population (25 runs) for a specific number of FEs. The results were averaged over all the runs for each generation. These results were then divided by the known optimum (best) solution for the particular problem. The values were normalized by dividing the previously obtained results with the maximum number. Thus, all of the values for each generation were normalized between 0 and 1. These values were used to plot the convergence graph, which varied with the generations (or FEs). We obtained the convergence values

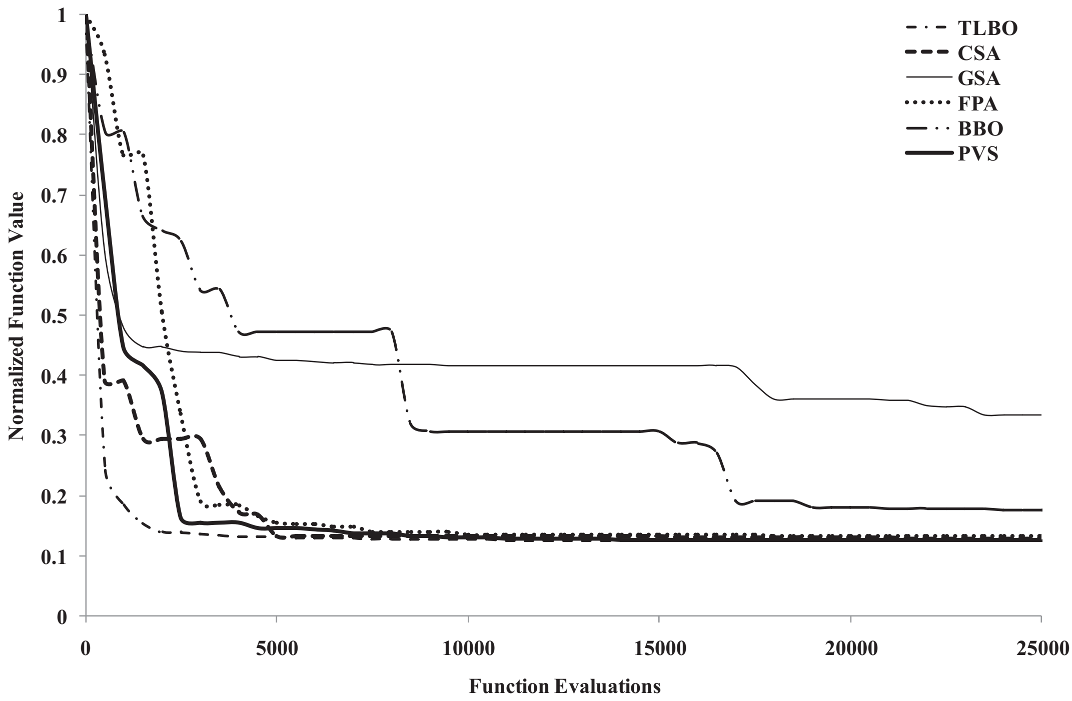

Fig. 9. Convergence of different meta-heuristics.

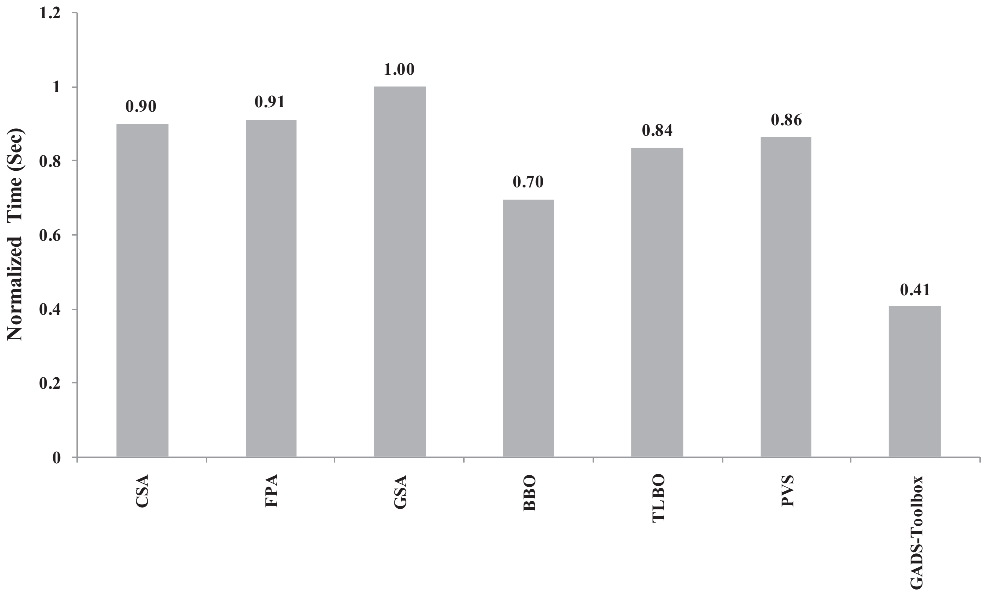

Fig. 10. Comparison of different meta-heuristics based on computational effort (normalized time).

using PVS, CSA, TLBO, FPA, GSA, and BBO for the step-cone, bearing, robot gripper, Belleville spring, welded stiffened shell, and planetary gear problems, because all of these problems had feasible solutions in each run. The convergence plot is given in Fig. 9 for 25,000 FEs with a population size of 50. Fig. 9 shows that the convergence was inferior using BBO and GSA compared with the other algorithms. During the initial generations, the convergence was better using TLBO than the other algorithms. The convergence was almost the same using PVS and CSA, which performed better than FPA. The result obtained by these algorithms converged as the number of generations increased, where the value was almost the same with all of the algorithms (about 10,000 FEs), except for BBO and GSA. We also checked the performance of the algorithms based on the computational effort required in terms of the time consumed running the algorithm under specific conditions. The time taken by each algorithm (PVS, TLBO, CSA, FPA, GSA, BBO, and GA-from Matlab GADS toolbox) for 25 runs was calculated for different engineering problems (step-cone, bearing, robot gripper, Belleville spring, welded stiffened shell, and planetary gear) using 25,000 FEs. These values were averaged for all of the problems and normalized to one with respect to the maximum value. These normalized time values are shown in Fig. 10, which demonstrate that GA (from Matlab GADS toolbox) was more computationally effective than the other algorithms, but this is expected because it is a commercially available tool. However, the GADS Toolbox obtained inferior results compared with the other algorithms. BBO was computationally effective and GSA was computationally poor compared with the other algorithms. The computational time was almost the same for TLBO and PVS, and they were better than CSA and FPA.

7. Conclusions

In this study, we proposed a new meta-heuristics optimization method called PVS. We showed that a "passing mechanism" can be implemented to generate an optimization algorithm. We applied this algorithm to 13 engineering design optimization problems and 13 challenging constrained benchmark functions. We evaluated the performance of PVS by considering various aspects such as the ability to find optimum solutions, the convergence of the algorithm, the computational effort, and time requirements. The effectiveness of PVS was demonstrated based on comparisons with the results obtained using many state-of-the-art meta-heuristics. PVS performed significantly better at the problems considered in this study, but we could not confirm that PVS is generally better than other meta-heuristics according to the No Free Lunch theorem. In future research using this algorithm, it will be necessary to check its performance and suitability for a particular class of problems. However, the application of PVS to engineering design problems and constrained benchmark problems indicate that the "passing vehicle" theory can be applied successfully to practical optimization problems. Thus, an aim of this study is to facilitate future research based on the proposed theory. Many theories are available for vehicle passing behavior on a two-lane highway by considering different approaches, which might be applied in a simplified model of passing vehicles. It is also necessary to reduce the computational time requirements for any meta-heuristics and the results of the present study demonstrate that PVS obtained good performance in terms of computational time. It would be useful to make PVS more computationally effective in future research.

Appendix. Engineering design problem

A.1. Planetary gear

Minimize,

$$f(X) = max|i_k - i_{0k}|,\ k = \{1, 2, R\},$$

where

$$i_1 = N_6/N_4$$

$$i_{01} = 3.11$$

$$i_2 = \frac{N_6(N_1N_3 + N_2N_4)}{N_1N_3(N_6 - N_4)}$$

$$i_{02} = 1.84$$

$$i_R = -\left(\frac{N_2N_6}{N_1N_3}\right)$$

$$i_{0R} = -3.11$$

$$X = \{N_1, N_2, N_3, N_4, N_5, N_6, p, m_1, m_2\},$$

subject to:

$$g_1(X) = m_3(N_6 + 2.5) \le D_{\max}$$

$$g_2(X) = m_1(N_1 + N_2) + m_1(N_2 + 2) \le D_{\max}$$

$$g_3(X) = m_3(N_4 + N_5) + m_3(N_5 + 2) \le D_{\max}$$

$$g_4(X) = |m_1(N_1 + N_2) - m_3(N_6 - N_3)| \le m_1 + m_3$$

$$g_5(X) = (N_1 + N_2)\sin\left(\frac{\pi}{p}\right) - N_2 - 2 - \delta_{22} \ge 0$$

$$g_6(X) = (N_6 - N_3)\sin\left(\frac{\pi}{p}\right) - N_3 - 2 - \delta_{33} \ge 0$$

$$g_7(X) = (N_4 + N_5)\sin\left(\frac{\pi}{p}\right) - N_5 - 2 - \delta_{55} \ge 0$$

$$g_8(X) = (N_6 - N_3)^2 + (N_4 + N_5)^2 - 2(N_6 - N_3)(N_4 + N_5)\cos\left(\frac{2\pi}{p} - \beta\right) \ge (N_3 + N_5 + 2 + \delta_{35})^2,$$

where

$$\beta = \frac{\cos^{-1}((N_6 - N_3)^2 + (N_4 + N_5)^2 - (N_3 + N_5)^2)}{2(N_6 - N_3)(N_4 + N_5)}$$

$$g_9(X) = N_6 - 2N_3 - N_4 - 4 - 2\delta_{34} \ge 0$$

$$g_{10}(X) = N_6 - N_4 - 2N_5 - 4 - 2\delta_{56} \ge 0$$

$$h(X) = \frac{N_6 - N_4}{p} = integer,$$

where

$$D_{max} = 220,\ p = (3, 4, 5),\ m_1, m_3 = (1.75, 2.0, 2.25, 2.5, 2.75, 3.0),\ \delta_{22}, \delta_{33}, \delta_{55}, \delta_{35}, \delta_{56} = 0.5$$

$$17 \le N_1 \le 96,\ 14 \le N_2 \le 54,\ 14 \le N_3 \le 51,\ 17 \le N_4 \le 46,\ 14 \le N_5 \le 51,\ 48 \le N_6 \le 124,\ N_i = integer.$$

A.2. Welded stiffened shell

Minimize

$$f(X) = K_M + \sum^{K_{Fi}} + K_P$$

$$X = \{t, n_s, n_r, h_r, h\},$$

subject to:

$$g_1(X) = \sigma_e = \sqrt{\sigma_a^2 - \sigma_a\sigma_p + \sigma_p^2} \le \frac{f_{y1}}{\sqrt{1 + \lambda_s^4}}$$

$$g_2(X) = \sigma_e \le \frac{f_{y1}}{\sqrt{1 + \lambda_p^4}}$$

$$g_3(X) = A_{Rreq} \le A_R$$

$$g_4(X) = I_{Rreq} \le I_R$$

$$g_5(X) = n_s \le \frac{2\left(R - \frac{h_r}{2}\right)\pi}{b + 300},$$

where

$$K_M = k_{M1}5\rho V_1 + k_{M1}\rho n_r V_R + k_{M2}\rho n_s A_s L$$

$$K_{F0} = 5k_F\theta e^{\mu}$$

$$\mu = 6.8582513 - 4.527217t^{-0.5} + 0.009541996(2R)^{0.5}$$

$$K_{F1} = 5k_F(\theta\sqrt{\kappa\rho V_1} + (1.3)(0.1520E-3)t^{1.9358}\,2L_s,$$

$$\theta = 2,\ \kappa = 2$$

$$K_{F2} = k_F(\theta\sqrt{25\rho V_1} + (1.3)(0.1520E-3)t^{1.9358}4(2R\pi)$$

$$K_{F3} = n_r k_F\left(3\sqrt{3\rho V_R} + (1.3)(0.3394E-3)a_{wr}^2 4\pi(R-h_r)\right)$$

$$K_{F4} = k_F\left(3\sqrt{n_r+1}\rho(5V_1 + n_r V_R) + (1.3)(0.3394E-3)a_{wr}^2 n_r 4R\pi\right)$$

$$K_{F5} = k_F\left(3\sqrt{(n_r+n_s+1)\rho(5V_1 + n_r V_R + n_s A_s L)} + (1.3)(0.3394E-3)a_{ws}^2 n_s 2L\right)$$

$$K_P = k_p\left(2R\pi L + 2R\pi(L - n_r h_r) + 2n_r\pi h_r(R-h_r) + 4\pi n_r h_r\left(R - \frac{h_r}{2}\right) + n_s L(h_1+2b)\right)$$

$$V_1 = 2R\pi t L_{s,},$$

$$V_R = 2\pi\delta_r h_r^2(R-h_r) + 4\pi\delta_r h_r^2\left(R - \frac{h_r}{2}\right),$$

$$\sigma_a = \frac{N_F}{2R\pi t_e},$$

$$t_e = t + \frac{A_s}{S},$$

$$S = \frac{2R\pi}{n_s}$$

$$\sigma_p = \frac{p_F R}{t(1+\alpha)},$$

$$\alpha = \frac{A_R}{L_{e0}t},$$

$$L_{e0} = \min\left(L_r,\ L_{er} = 1.56\sqrt{Rt}\right),$$

$$L_r = \frac{L}{n_r - 1},$$

$$\sigma_e =$$

$$\lambda_s^2 = f_{y1}/\sigma_e\left(\frac{\sigma_a}{\sigma_{Eas}} + \frac{\sigma_p}{\sigma_{Eps}}\right)$$

$$f_{y1} = \frac{f_y}{1.1}$$

$$\sigma_{Eas} = c_{as}\frac{\pi^2 E}{12\left(1-\upsilon^2\right)}\left(\frac{t}{s}\right)^2,$$

$$c_{as} = \psi_{as}\sqrt{1 + \left(\frac{\rho_{as}\xi_{as}}{\psi_{as}}\right)^2},$$

$$z_{as} = \frac{s^2}{Rt}\sqrt{1-\upsilon^2},$$

$$\xi_{as} = 0.702 z_{as}$$

$$\rho_{as} = 0.5\left(1 + \frac{R}{150t}\right)^{-0.5},$$

$$\sigma_{Eps} = \frac{c_{ps}\pi^2 E}{10.92}\left(\frac{t}{s}\right)^2,$$

$$c_{ps} = \psi_{ps}\sqrt{1 + \left(\frac{\rho_{ps}\xi_{ps}}{\psi_{ps}}\right)^2},$$

$$\xi_{ps} = 1.04\left(\frac{s}{L_r}\right)\sqrt{z_{ps}},$$

$$z_{ps} = z_{as},$$

$$\psi_{ps} = \left(1 + \left(\frac{s}{L_r}\right)^2\right)^2,$$

$$\lambda_p^2 = \frac{f_{y1}}{\sigma_e\left(\frac{\sigma_a}{\sigma_{Eap}} + \frac{\sigma_p}{\sigma_{Epp}}\right)},$$

$$\sigma_{Eap} = \left(\frac{c_{ap}\pi^2 E}{10.92}\right)\left(\frac{t}{L_r}\right)^2,$$

$$c_{ap} = \psi_{ap}\sqrt{1 + \left(\frac{\rho_{ap}\xi_{ap}}{\psi_{ap}}\right)^2},$$

$$\xi_{ap} = 0.702 z_{ap},$$

$$z_{ap} = 0.9539\frac{L_r^2}{Rt}$$

$$z_{pp} = z_{ap},$$

$$\psi_{ap} = \frac{1 + \gamma_s}{1 + \frac{A_s}{s_e t}},$$

$$\gamma_s = \frac{10.92 I_{sef}}{st^3},$$

$$s_E = 1.9t\sqrt{E/f_y},$$

$$if\ s_E \le s \to s_e = s_E,\ else\ s_e = s$$

$$Z_G = \frac{\frac{h_1}{2}t_w\left(\frac{h_1}{4} + \frac{t}{2}\right) + bt_f\left(\frac{h_1+t+t_f}{2}\right)}{s_e t + bt_f + \frac{h_1 t_w}{2}}$$

$$I_{sef} = s_e t z_G^2 + \left(\frac{h_1}{2}\right)^3\left(\frac{t_w}{12}\right) + \frac{h_1 t_w}{2}\left(\frac{h_1}{4} + \frac{t}{2} - z_G\right)^2 + bt_f\left(\frac{h_1 + t + t_f}{2} - z_G\right)^2,$$

$$A_s = bt_f + \frac{h_1 t_w}{2}$$

$$\sigma_{Epp} = \frac{c_{pp}\left(\pi^2 E\right)}{10.92}\left(\frac{t}{L_r}\right)^2,$$

$$c_{pp} = \psi_{pp}\sqrt{1 + \left(\frac{\rho_{pp}\xi_{pp}}{\psi_{pp}}\right)^2},$$

$$\xi_{pp} = 1.04\sqrt{z_{pp},}$$

$$\psi_{pp} = 2\left(1 + \sqrt{1 + \gamma_s}\right),$$

$$t_r = \delta_r h_r,$$

$$\frac{1}{\delta_r} = 42\varepsilon,$$

$$\varepsilon = \sqrt{235/f_y},$$

$$\delta_r = \frac{1}{34.17189},$$

$$A_R = 3h_r t_r = 3\delta_r h_r^2,$$

$$A_{Rreq} = \left(\frac{2}{Z^2} + 0.06\right)L_r t,$$

$$Z = 0.9539\frac{L_r^2}{Rt},$$

$$L_e = \min\left(L_r,\ 2\left(1.56\sqrt{Rt}\right)\right)$$

$$y_E = \frac{L_e t\left(h_r + \frac{t}{2}\right) + \delta_r h_r^3}{3\delta_r h_r^2 + L_e t},$$

$$I_R = \frac{\delta_r h_r^4}{6} + 2\delta_r h_r^2\left(\frac{h_r}{2} - y_E\right)^2 + \delta_r h_r^2 y_E^2 + L_e t\left(h_r + \frac{t}{2} - y_E\right)^2,$$

$$I_{Rreq} = I_a + I_p,$$

$$I_a = \frac{\sigma_a t\left(1 + \frac{A_s}{st}\right)R_0^4}{500EL_r},$$

$$R_0 = R - (h_r - y_E),$$

$$I_p = \frac{p_F R R_0^2 L_r}{3E}\left(2 + \frac{3Ey_E\delta_0}{R_0^2\left(\frac{f_y}{2} - \sigma_p\right)}\right),$$

$$\delta_0 = 0.005R,$$

$$a_{ws} = 0.4t_w,$$

$$a_{wr} = 0.4t_r$$

$$t_f = \sqrt{33.20534 + (6.701288E-4)h^2}$$

$$b = \sqrt{5851.785 + (1.671844E-2)h^2\log(h)}$$

$$t_w = \sqrt{15.62577 + (4.358947E-5)h^2 \log(h)}$$

$$h_1 = h - 2t_f$$

$$L_s = 3000,\ \rho = 7.85E-6,\ N_f = 5.4E7,\ p_F = 1.5,\ L = 15000,\ R = 1850,\ f_y = 355,\ E = 2.1E5,\ \upsilon = 0.3,\ k_{m1} = k_{m2} = k_f = 1,\ k_p = 14.4E-6,\ \rho_{ps} = 0.6,\ \psi_{as} = 4,\ \rho_{ap} = 0.5,\ \rho_{pp} = 0.6,$$

$$4 \le t, n_s, n_r \le 40,$$

$$130 \le h_r \le 510,$$

$$h = (152, 203, 254, 305, 356, 406, 457,\ 533, 610, 686, 762, 838,\ \text{and}\ 914),$$

$$t, n_s, n_r = integer, h_r \text{ varies in a step of } 10.$$

A.3. 4-stage gear box

Minimize

$$f = \left(\frac{\pi}{1000}\right)\sum_{i=1}^{4}\frac{b_i c_i^2\left(N_{pi}^2 + N_{gi}^2\right)}{\left(N_{pi} + N_{gi}\right)^2},$$

where $i = (1, 2, 3, 4)$
subject to:

$$g_1 = \left(\frac{366000}{\pi\omega_1} + \frac{2c_1N_{p1}}{N_{p1}+N_{g1}}\right)\left(\frac{\left(N_{p1}+N_{g1}\right)^2}{4b_1c_1^2N_{p1}}\right) \le \frac{\sigma_N J_R}{0.0167WK_oK_m}$$

$$g_2 = \left(\frac{366000N_{g1}}{\pi\omega_1N_{p1}} + \frac{2c_2N_{p2}}{N_{p2}+N_{g2}}\right)\left(\frac{\left(N_{p2}+N_{g2}\right)^2}{4b_2c_2^2N_{p2}}\right) \le \frac{\sigma_N J_R}{0.0167WK_oK_m}$$

$$g_3 = \left(\frac{366000N_{g1}N_{g2}}{\pi\omega_1N_{p1}N_{p2}} + \frac{2c_3N_{p3}}{N_{p3}+N_{g3}}\right)\left(\frac{\left(N_{p3}+N_{g3}\right)^2}{4b_3c_3^2N_{p3}}\right) \le \frac{\sigma_N J_R}{0.0167WK_oK_m}$$

$$g_4 = \left(\frac{366000N_{g1}N_{g2}N_{g3}}{\pi\omega_1N_{p1}N_{p2}N_{p3}} + \frac{2c_4N_{p4}}{N_{p4}+N_{g4}}\right)\left(\frac{\left(N_{p4}+N_{g4}\right)^2}{4b_4c_4^2N_{p4}}\right) \le \frac{\sigma_N J_R}{0.0167WK_oK_m}$$

$$g_5 = \left(\frac{366000}{\pi\omega_1} + \frac{2c_1N_{p1}}{N_{p1}+N_{g1}}\right)\left(\frac{\left(N_{p1}+N_{g1}\right)^3}{4b_1c_1^2N_{g1}N_{p1}^2}\right) \le \left(\frac{\sigma_H}{C_p}\right)^2\left(\frac{sin\varphi cos\varphi}{0.0334WK_oK_m}\right)$$

$$g_6 = \left(\frac{366000N_{g1}}{\pi\omega_1N_{p1}} + \frac{2c_2N_{p2}}{N_{p2}+N_{g2}}\right)\left(\frac{\left(N_{p2}+N_{g2}\right)^3}{4b_2c_2^2N_{g2}N_{p2}^2}\right) \le \left(\frac{\sigma_H}{C_p}\right)^2\left(\frac{sin\varphi cos\varphi}{0.0334WK_oK_m}\right)$$

$$g_7 = \left(\frac{366000N_{g1}N_{g2}}{\pi\omega_1N_{p1}N_{p2}} + \frac{2c_3N_{p3}}{N_{p3}+N_{g3}}\right)\left(\frac{\left(N_{p3}+N_{g3}\right)^3}{4b_3c_3^2N_{g3}N_{p3}^2}\right) \le \left(\frac{\sigma_H}{C_p}\right)^2\left(\frac{sin\varphi cos\varphi}{0.0334WK_oK_m}\right)$$

$$g_8 = \left(\frac{366000N_{g1}N_{g2}N_{g3}}{\pi\omega_1N_{p1}N_{p2}N_{p3}} + \frac{2c_4N_{p4}}{N_{p4}+N_{g4}}\right)\left(\frac{\left(N_{p4}+N_{g4}\right)^3}{4b_4c_4^2N_{g4}N_{p4}^2}\right) \le \left(\frac{\sigma_H}{C_p}\right)^2\left(\frac{sin\varphi cos\varphi}{0.0334WK_oK_m}\right)$$

$$g_{9-12} = N_{pi}\sqrt{\frac{sin^2\varphi}{4} + \frac{1}{N_{pi}} + \left(\frac{1}{N_{pi}}\right)^2} + N_{gi}\sqrt{\frac{sin^2\varphi}{4} + \frac{1}{N_{gi}} + \left(\frac{1}{N_{gi}}\right)^2} - \frac{sin\varphi\left(N_{pi}+N_{gi}\right)}{2} \ge CR_{\min}\ \pi cos\varphi$$

$$g_{13-16} = d_{\min} \le \frac{2c_i N_{pi}}{N_{pi} + N_{gi}}$$

$$g_{17-20} = d_{\min} \le \frac{2c_i N_{gi}}{N_{pi} + N_{gi}}$$

$$g_{21} = x_{p1} + \left(\frac{\left(N_{p1} + 2\right)c_1}{N_{p1} + N_{g1}} \right) \le L_{\max}$$

$$g_{22-24} = (x_{g(i-1)} + \left(\frac{\left(N_{pi} + 2\right)c_i}{N_{pi} + N_{gi}} \right)_{i=2,3,4} \le L_{\max}$$

$$g_{25} = -x_{p1} + \frac{\left(N_{p1} + 2\right)c_1}{N_{p1} + N_{g1}} \le 0$$

$$g_{26-28} = \left(-x_{g(i-1)} + \frac{\left(N_{pi} + 2\right)c_i}{N_{pi} + N_{gi}} \right)_{i=2,3,4} \le 0$$

$$g_{29} = y_{p1} + \frac{\left(N_{p1} + 2\right)c_1}{N_{p1} + N_{g1}} \le L_{\max}$$

$$g_{30-32} = \left(y_{g(i-1)} + \frac{\left(N_{pi} + 2\right)c_i}{N_{pi} + N_{gi}} \right)_{i=2,3,4} \le L_{\max}$$

$$g_{33} = -y_{p1} + \frac{\left(N_{p1} + 2\right)c_1}{N_{p1} + N_{g1}} \le 0$$

$$g_{34-36} = \left(-y_{g(i-1)} + \frac{\left(N_{pi} + 2\right)c_i}{N_{pi} + N_{gi}} \right)_{i=2,3,4} \le 0$$

$$g_{37-40} = x_{gi} + \frac{\left(N_{gi} + 2\right)c_i}{N_{pi} + N_{gi}} \le L_{\max}$$

$$g_{41-44} = (-x_{gi} + \left(\frac{\left(N_{gi} + 2\right)c_i}{N_{pi} + N_{gi}} \right) \le 0$$

$$g_{45-48} = (y_{gi} + \left(\frac{\left(N_{gi} + 2\right)c_i}{N_{pi} + N_{gi}} \right) \le L_{\max}$$

$$g_{49-52} = (-y_{gi} + \left(\frac{\left(N_{gi} + 2\right)c_i}{N_{pi} + N_{gi}} \right) \le 0$$

$$g_{53-56} = \left(0.945c_i - N_{pi} - N_{gi}\right)(b_i - 5.715)(b_i - 8.255)(b_i - 12.70)(-1) \le 0$$

$$g_{57-60} = \left(0.646c_i - N_{pi} - N_{gi}\right)(b_i - 3.175)(b_i - 8.255)(b_i - 12.70)(+1) \le 0$$

$$g_{61-64} = \left(0.504c_i - N_{pi} - N_{gi}\right)(b_i - 3.175)(b_i - 5.715)(b_i - 12.70)(-1) \le 0$$

$$g_{65-68} = \left(0c_i - N_{pi} - N_{gi}\right)(b_i - 3.175)(b_i - 5.715)(b_i - 8.255)(+1) \le 0$$

$$g_{69-72} = \left(N_{pi} + N_{gi} - 1.812c_i\right)(b_i - 5.715)(b_i - 8.255)(b_i - 12.70)(-1) \le 0$$

$$g_{73-76} = \left(N_{pi} + N_{gi} - 0.945c_i\right)(b_i - 3.175)(b_i - 8.255)(b_i - 12.70)(+1) \le 0$$

$$g_{77-80} = \left(N_{pi} + N_{gi} - 0.646c_i\right)(b_i - 3.175)(b_i - 5.715)(b_i - 12.70)(-1) \le 0$$

$$g_{81-84} = \left(N_{pi} + N_{gi} - 0.504c_i\right)(b_i - 3.175)(b_i - 5.715)(b_i - 8.255)(+1) \le 0$$

$$g_{85} = \omega_{\min} \le \frac{\omega_1\left(N_{p1}N_{p2}N_{p3}N_{p4}\right)}{\left(N_{g1}N_{g2}N_{g3}N_{g4}\right)}$$

$$g_{86} = \frac{\omega_1\left(N_{p1}N_{p2}N_{p3}N_{p4}\right)}{\left(N_{g1}N_{g2}N_{g3}N_{g4}\right)} \le \omega_{\max} ,$$

where

$$c_i = \sqrt{(x_{gi} - x_{pi})^2 + (y_{gi} - y_{pi})^2}$$

$$CR_{min} = 1.4, d_{min} = 25.4, \emptyset = 20^o, W = 55.9, J_R = 0.2, K_M = 1.6, K_0 = 1.5, L_{max} = 127, \sigma_H = 3290, \sigma_N = 2090, \omega_1 = 5000, \omega_{min} = 245, \omega_{max} = 255, C_p = 464,$$

$$x_{p1}, y_{p1}, x_{gi}, y_{gi} = (12.7,\ 25.4,\ 38.1,\ 50.8,\ 63.5,\ 76.2,\ 88.9,\ 101.6,\ 114.3)$$

$$b_i = (3.175,\ 5.715,\ 8.255,\ 12.7)$$

$$7 \le N_{pi},\ N_{gi} \le 76,\ N_{pi}, N_{gi} = integer$$

$$X = \{N_{p1}, N_{g1}, N_{p2}, N_{g2} \ldots .b_1, b_2 \ldots .x_{p1}, x_{g1}, x_{g2} \ldots y_{p1}, y_{g1}, y_{g2} \ldots y_{g4}\}.$$

References

[1] B. Xing, W.J. Gao, Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms. Intelligent Systems Reference Library 62, Springer International Publishing, Switzerland, 2014, doi:10.1007/978-3-319-03404-1.
[2] J. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, MI, 1975.
[3] M. Dorigo, Optimization, Learning and Natural Algorithms, Politecnico di Milano, Italy, 1992 PhD thesis.
[4] V. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of the IEEE International Conference on Neural Networks, 1995, pp. 1942–1948.
[5] J.D. Farmer, N. Packard, A. Perelson, The immune system, adaptation and machine learning, Phys. D 2 (1986) 187–204.
[6] Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. Technical report-TR06. Erciyes University, Engineering Faculty, Computer Engineering Department.
[7] D. Simon, Biogeography-based optimization, IEEE Trans. Evolut. Comput. 12 (2008) 702–713.
[8] K.M. Passino, Biomimicry of bacterial foraging for distributed optimization and control, IEEE Control Syst. Manag. 22 (2002) 52–67.
[9] X.S. Yang, A new metaheuristic bat-inspired algorithm, Nature inspired cooperative strategies for optimization (NISCO 2010), Studies in Computational Intelligence, Springer, Berlin, 2010, pp. 65–74. SCI 284.
[10] S.C. Chu, P.W. Tsai, Computational intelligence based on the behavior of cats, Int. J. Innovative Comput., Inf. Control 3 (2007) 163–173.
[11] X.S. Yang, S. Deb, Cuckoo search via Lévy flights, in: Proceedings of the World Congress on Nature and Biologically Inspired Computing (NaBIC), India, 2009, pp. 210–214.
[12] K.N. Krishnanand, D. Ghose, Detection of multiple source locations using a glowworm metaphor with applications to collective robotics, in: Proceedings of the IEEE Swarm Intelligence Symposium (SIS), 2005, pp. 84–91.
[13] X.S. Yang, Firefly algorithms for multimodal optimization, in: Stochastic algorithms: foundations and applications, Springer, Berlin Heidelberg, 2009, pp. 169–178.
[14] M. Hersovici, M. Jacovi, Y.S. Maarek, D. Pelleg, M. Shtalhaim, S. Ur, The shark search algorithm. An application: tailored Web site mapping, Comput. Netw. ISDN Syst. 30 (1998) 317–326.
[15] X.L. Li, J.X. Qian, Studies on artificial fish swarm optimization algorithm based on decomposition and coordination techniques, J. Circuits Syst. 8 (2003) 1–6.
[16] M.M. Eusuff, K.E. Lansey, Optimization of water distribution network design using the shuffled frog leaping algorithm, J. Water Resour. Plan. Manag. 129 (2003) 210–225.
[17] M. Taherdangkoo, M.H. Shirzadi, M.H. Bagheri, A novel meta-heuristic algorithm for numerical function optimization_blind, naked mole-rats (BNMR) algorithm, Sci. Res. Essays 7 (2012) 3566–3583.
[18] Z. Chen, H. Tang, Cockroach swarm optimization, in: Proceedings of the IEEE 2nd International Conference on Computer Engineering and Technology (ICCET), 2010, pp. 652–655.
[19] M.C. Su, S.Y. Su, Y.X. Zhao, A swarm-inspired projection algorithm, Pattern Recognit. 42 (2009) 2764–2786.
[20] X.S. Yang, S. Deb, Eagle strategy using Lévy walk and firefly algorithms for stochastic optimization, in: J.R. Gonzalez (Ed.), Nature Inspired Cooperative Strategies for Optimization (NISCO 2010), SCI, 284, Springer, Berlin, 2010, pp. 101–111.
[21] J.Y. Liu, M.Z. Guo, C. Deng, Geese PSO: An efficient improvement to particle swarm optimization, Comput. Sci. 33 (2006) 166–168.
[22] A. Mucherino, O. Seref, Monkey search: A novel metaheuristic search for global optimization, in: Proceedings of AIP Conference, 953, 2007, pp. 162–173.
[23] C. Liu, X. Yan, C. Liu, H. Wu, The wolf colony algorithm and its application, Chin. J. Electron. 20 (2011) 212–216.
[24] A.R. Mehrabian, C. Lucas, A novel numerical optimization algorithm inspired from weed colonization, Ecol. Inf. 1 (2006) 355–366.
[25] X.S. Yang, Flower pollination algorithm for global optimization, Unconventional Computation and Natural Computation,, 7445, Springer, Berlin, 2012, pp. 240–249. LNCS.

[26] A. Kaveh, S. Talatahari, A novel heuristic optimization method: charged system search, Acta Mech. 213 (2010) 267–289.
[27] S.I. Birbil, S.C. Fang, An electromagnetism-like mechanism for global optimization, J. Global Optim. 25 (2003) 263–282.
[28] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: A gravitational search algorithm, Inf. Sci. 179 (2009) 2232–2248.
[29] H. Shah-Hosseini, Problem solving by intelligent water drops, in: Proceedings of the IEEE Congress on Evolutionary Computation (CEC-2007), 2007, pp. 3226–3231.
[30] H. Eskandar, A. Sadollah, A. Bahreininejad, M. Hamdi, Water cycle algorithm: A novel meta-heuristic optimization for solving constrained engineering optimization problems, Comput. Struct. 110–111 (2012) 151–166.
[31] Z.W. Geem, J.H. Kim, G.V. Loganathan, A new heuristic optimization algorithm: Harmony search, Simulation 76 (2001) 60–68.
[32] E. Atashpaz-Gargari, C. Lucas, Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition, in: Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2007), 2007, pp. 4661–4667.
[33] Y. Cui, R. Guo, D. Guo, A naïve five-element string algorithm, J. Softw. 4 (2009) 925–934.
[34] R.V. Rao, D.P. Vakharia, V.J. Savsani, Mechanical engineering design optimization using modified harmony elements algorithm, Int. J. Des. Eng. 2 (2009) 116–135.
[35] A. Ahrari, M. Shariat-Panahi, A.A. Atai, GEM: a novel evolutionary optimization method with improved neighborhood search, Appl. Math. Comput. 210 (2009) 379–386.
[36] R.V. Rao, V.J. Savsani, D.P. Vakharia, Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems, Comput.-Aided Des 43 (2011) 303–315.
[37] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evolut. Comput. 1 (1) (1997) 67–82.
[38] Y. Wang, Z. Cai, Q. Zhang, Differential evolution with composite trial vector generation strategies and control parameters, IEEE Trans. Evolut. Comput. 15 (1) (2011) 55–66.
[39] Y. Wang, Z. Cai, Q. Zhang, Enhancing the search ability of differential evolution through orthogonal crossover, Inf. Sci. 185 (1) (2012) 153–177.
[40] Y. Wang, H.X. Li, T. Huang, L. Li, Differential evolution based on covariance matrix learning and bimodal distribution parameter setting, Appl. Soft Comput. 18 (2014) 232–247.
[41] D. Joaquín, G. Salvador, M. Daniel, H. Francisco, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evolut. Comput. 1 (1) (2011) 3–18.
[42] S. Holm, A simple sequentially rejective multiple test procedure, Scandinavian J. Stat. 6 (2) (1979) 65–70.
[43] R. Lamm, A. Beck, T. Ruscher, T. Mailaender, S. Cafiso, G. La Cava, W. Matthews, How to Make Two-Lane rural Roads Safer-Scientific Background and Guide for Practical Application, WIT Press., 2006.
[44] W. Brilon, F. Weiser, Capacity and speed-flow relationships on rural two-lane highways in Germany, in: Proceedings of the Third International Symposium on Highway Capacity: Country Reports, Copenhagen: Road Directorate, Transportation Research Board, 1998, pp. 199–218.
[45] S. Erlander, A mathematical model for traffic on a two-lane road with some empirical results—I. Theoretical model and estimation problems, Transp. Res. 5 (2) (1971) 135–147.
[46] S. Erlander, A mathematical model for traffic on a two-lane road with some empirical results—II. Empirical results., Transp. Res. 5 (2) (1971) 149–175.
[47] A.H. Ghods, F. Saccomanno, Microscopic overtaking gap acceptance for two-lane highways, in: Proceedings of the Transportation Research Board 93rd Annual Meeting (No. 14-3374),, Washington D.C., 2013.
[48] H.F. Hammond, L.J. Sorenson, Traffic Engineering Handbook, Institute of Traffic Engineers and National Conservation Bureau, New York, 1941.
[49] Kallberg, H. (1980). Overtakings and Platoons on Two-Lane Rural Roads. (Report 61). Technical Research Center of Finland, Road and Traffic Laboratory.
[50] T. Luttinen, Level of service on Finnish two-lane highways, Proceedings of the Transportation Research Circular E-C018: Fourth International Symposium on Highway Capacity, 2002, pp. 175–187.
[51] J.R. McLean, Two-Lane Highway Traffic Operations: Theory and Practice, Gordon and Breach Science Publishers, New York, 1989.
[52] A.J. Miller, Analysis of bunching in rural two-lane traffic, Oper. Res. 11 (1963) 236–247.
[53] G.F. Yeo, Traffic delays on a two-lane road, Biometrika 51 (1964) 11–15.
[54] T. Luttinen, Uncertainty in operational analysis of two-lane highways, Transp. Res. Record: J. Transp. Res. Board 1802 (1) (2002) 105–114.
[55] T. Luttinen, Percent time-spent-following as performance measure for two-lane highways, Transp. Res. Record: J. Transp. Res. Board 1776 (1) (2001) 52–59.
[56] Luttinen, T. (2001). Traffic flow on two-lane highways: an overview. TL Consulting Engineers-Research Report.
[57] S.S. Rao, Engineering Optimization: Theory and Practice, New Age International, New Delhi, 2002.
[58] Y. Wang, Z. Cai, Y. Zhou, Z. Fan, Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique, Struct. Multidiscip. Optim. 37 (4) (2009) 395–413.
[59] B. Adil, B.O. Fehmi, Adaptive firefly algorithm with chaos for mechanical design optimization problems, Appl. Soft Comput. 36 (2015) 152–164.
[60] B. Akay, D. Karaboga, Artificial bee colony algorithm for large-scale problems and engineering design optimization, J. Intell. Manuf. 23 (4) (2012) 1001–1014.
[61] S. Chun, Y.T. Kim, D.T.-H. Kim, A diversity-enhanced constrained particle swarm optimizer for mixed integer-discrete-continuous engineering design problems, Adv. Mech. Eng. 5 (2013) 130750 http://dx.doi.org/10.1155/2013/130750 .
[62] A.H. Gandomi, X.S. Yang, A.H. Alavi, S. Talatahari, Bat algorithm for constrained optimization tasks, Neural Comput. Appl. 22 (6) (2013) 1239–1255.
[63] A.H. Gandomi, X.S. Yang, A.H. Alavi, S. Talatahari, Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems, Eng. Comput. 29 (2013) 17–35.
[64] A.H. Gandomi, Interior search algorithm (ISA): a novel approach for global optimization, ISA Trans. 53 (2014) 1168–1183.
[65] H. Liu, Z. Cai, Y. Wang, Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization, Appl. Soft Comput. 10 (2) (2010) 629–640.
[66] I. Mazhoud, K. Hadj-Hamou, J. Bigeon, P. Joyeux, Particle swarm optimization for solving engineering problems: a new constraint-handling mechanism, Eng. Appl. Artif. Intell. 26 (4) (2013) 1263–1273.
[67] M. Montemurro, A. Vincenti, P. Vannucci, The automatic dynamic penalisation method (ADP) for handling constraints with genetic algorithms, Comput. Methods Appl. Mech. Eng. 256 (2013) 70–87.
[68] E. Zahara, Y.T. Kao, Hybrid Nelder–Mead simplex search and particle swarm optimization for constrained engineering design problems, Expert Syst. Appl. 36 (2009) 3880–3886.
[69] S. Ali, B. Ardeshir, E. Hadi, H. Mohd, Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems, Appl. Soft Comput. 13 (2013) 2592–2612.
[70] P.A. Simionescu, D. Beale, G.V. Dozier, Teeth-number synthesis of a multispeed planetary transmission using an estimation of distribution algorithm, ASME J. Mech. Des. 128 (2006) 108–115.
[71] K. Jarmai, J.A. Snyman, J. Farkas, Minimum cost design of a welded orthogonally stiffened cylindrical shell, Comput. Struct. 84 (2006) 787–797.
[72] L.P. Pomrehn, P.Y. Papalambros, Discrete optimal design formulations with application to gear train design, J. Mech. Des. 117 (3) (1995) 419–424.
[73] R.V. Rao, V.J. Savsani, Mechanical Design Optimization Using Advanced Optimization Techniques, Springer, London, 2012 ISBN 978-1-4471-2747-5.
[74] G. Lechner, H. Naunheimer, Automotive Transmissions: Fundamentals, Selection, Design, and Application, Springer, New York, 1999.
[75] L. Wang, L.P. Li, An effective differential evolution with level comparison for constrained engineering design, Struct. Multidiscip.Optim. 41 (2010) 947–963.
[76] S. Gupta, R. Tiwari, B.N. Shivashankar, Multi-objective design optimization of rolling bearings using genetic algorithm, Mech. Mach. Theory 42 (2007) 1418–1443.
[77] G. Wenyin, C. Zhihua, L. Dingwen, Engineering optimization by means of an improved constrained differential evolution, Comput. Methods Appl. Mech. Eng. 268 (2014) 884–904.
[78] T. Luttinen, Percent time-spent-following as performance measure for two-lane highways, Trans. Res. Record: J. Trans. Res. Board 1776 (1) (2001a) 52–59.
[79] T. Luttinen, Traffic flow on two-lane highways: an overview. TL Consulting Engineers-Research Report, 2001b.