Structural and Multidisciplinary Optimization (2018) 58:1505–1522
https://doi.org/10.1007/s00158-018-1978-3

RESEARCH PAPER

A parallel boundary search particle swarm optimization algorithm for constrained optimization problems

Zhao Liu [1] · Zeyang Li [1] · Ping Zhu [1] · Wei Chen [1,2]

Received: 11 December 2017 / Revised: 16 March 2018 / Accepted: 19 March 2018 / Published online: 13 April 2018

Abstract

During the past decade, considerable research has been conducted on constrained optimization problems (COPs) which are frequently encountered in practical engineering applications. By introducing resource limitations as constraints, the optimal solutions in COPs are generally located on boundaries of feasible design space, which leads to search difficulties when applying conventional optimization algorithms, especially for complex constraint problems. Even though penalty function method has been frequently used for handling the constraints, the adjustment of control parameters is often complicated and involves a trial-and-error approach. To overcome these difficulties, a modified particle swarm optimization (PSO) algorithm named parallel boundary search particle swarm optimization (PBSPSO) algorithm is proposed in this paper. Modified constrained PSO algorithm is adopted to conduct global search in one branch while Subset Constrained Boundary Narrower (SCBN) function and sequential quadratic programming (SQP) are applied to perform local boundary search in another branch. A cooperative mechanism of the two branches has been built in which locations of the particles near boundaries of constraints are selected as initial positions of local boundary search and the solutions of local boundary search will lead the global search direction to boundaries of active constraints. The cooperation behavior of the two branches effectively reinforces the optimization capability of the PSO algorithm. The optimization performance of PBSPSO algorithm is illustrated through 13 CEC06 test functions and 5 common engineering problems. The results are compared with other state-of-the-art algorithms and it is shown that the proposed algorithm possesses a competitive global search capability and is effective for constrained optimization problems in engineering applications.

Keywords Particle swarm optimization · Constrained optimization problems · Subset constraints boundary narrower function · Parallel boundary search · Diversity enhancement

1 Introduction

Constrained optimization problems (COPs) are the most popular type of problems in engineering optimization. COPs are generally composed of objective function, constrained functions, variables and bounds (search space) and can be formulated as (1):

$$\begin{aligned} &Find\ \ x{\in}S{\subseteq}R^{D} \\ &\min\ f\left(\overrightarrow{x}\right) \\ &s.t.\quad g_i{\le}0,\ \ for\ \ i=1\ \ to\ \ q \\ &\qquad\ \ g_i=0,\ \ for\ \ i=q+1\ \ to\ \ m \end{aligned} \tag{1}$$

Here *f* and *g* are objective function and constrained functions, respectively. $\overrightarrow{\mathrm{x}}$ are variables and S represents design space defined on R^{D}. *q* is the number of inequalities and $m-q$ is equalities constrained functions.

✉ Ping Zhu
pzhu@sjtu.edu.cn

[1] The State key laboratory of Mechanical System and Vibration, School of Mechanical Engineering, Shanghai Jiao Tong University, 800 DongChuan RD, Shanghai 200240, People's Republic of China

[2] Department of Mechanical Engineering, Northwestern University, 2145 Sheridan RD Tech B224, Evanston, IL 60201, USA

For practical engineering optimization problems, the objective function value is generally an engineering objective such as the minimization of cost or mass. The resource limitation is often expressed as constraint. When an optimal solution sits on a boundary of feasible design space, it implies resource is exhausted. By agreement, boundary of constraint refers to as boundary of feasible design space in this article. At boundaries of constraints, values of constraint become zero and these constraints are called active constraints (Bonyadi and Michalewicz 2014). Many existing optimization algorithms have difficulties in handling constraints. Taking penalty function method as an example, a mild punishment may lead to infeasible solutions while excessive punishment may restrict particles moving to regions near boundaries of constraints and therefore, optimal solutions are likely to be missed.

In this paper, one of the existing global optimization algorithms, Particle Swarm Optimization algorithm (PSO), is extended to solve the constrained optimization problem combined with a novel local boundary search method. PSO algorithm was originally proposed by Kennedy and Eberhart (Kennedy and Eberhart 1995). Imitating the cooperative behaviors of fish schooling and bird flocking, the algorithm is based on particle swarm to converge to an optimal solution. Two characteristics of a particle include position and velocity guide motivation. Position is the current location in a search region and velocity is updated according to the individual and the global search history.

While the concept is simple, the PSO algorithm has a strong search capability and becomes a very popular global optimization algorithm. It also has been extended to solve COPs. Parsopoulos (Parsopoulos and Vrahatis 2002) incorporated the non-stationary multi-stage penalty function method into PSO algorithm and the testing results demonstrated the advantage of PSO for solving COPs. Pulido (Pulido and Coello 2004) proposed the feasibility-based rules method for handling constraints in which the global best of particle swarm was chosen based on feasibility rules. A turbulence operator was also used in this method to improve the exploratory ability. Hu (Hu and Eberhart 2002) and Guo (Guo et al. 2004) used a fly-back mechanism to keep particles from escaping the feasible region so that the feasibility of solutions can be proved. He (He and Wang 2007a) proposed a co-evolution PSO approach to solve COPs. In that method, along with the optimization process, the penalty function evolves in a self-adaptive way. He and Wang (He and Wang 2007b) employed the simulated annealing method with PSO algorithm to improve the global search capability of the standard PSO and the feasibility-based rules were used for handling the constrain functions. Paquet (Paquet and Engelbrecht 2007) proposed a PSO variant to solve linear equality constrained problems. All particles positions are initialized at the position satisfying the constraints so that all further generated particles are feasible. Liang (Liang et al. 2010) proposed a CoCLPSO method to deal with COPs in which particles were adaptively assigned to explore different constraints according to their difficulties and the SQP method is also combined in CoCLPSO to improve its local search ability. Sun (C-l et al. 2011) proposed the IVPSO to update the position of particles in a vector pattern and the one-dimensional search method was applied to find feasible positions for particles escaping from the feasible region. Bonyadi (Bonyadi et al. 2013) proposed a constraint handling method in cooperation with MLPSO method to solve COPs. It was assumed that any particles with feasible positions are better than infeasible particles and infeasible particles with smaller constraint violation values are better than those with larger values. Mezura-Montes and Coello (Mezura-Montes and Coello Coello 2011) published a survey paper summarizing considerable amount of existing constraint-handling techniques within nature-inspired algorithms. Main features of each constraint-handling technique have been given and it should be noted that extra parameters are generally required in most of the methods.

As stated above, present researches on constrained PSO algorithms focus on constraint handling mechanism to keep solution feasible while ignoring the fact that a global optimum generally locates near boundaries of constraints. Also, to improve the accuracy of constraint handling, control parameters are often used in these algorithms but adjusting parameters for different problems is always complicated and time-consuming. To overcome the limitations of existing methods, our goal in this work is to improve the constrained PSO algorithm by enhancing local search near boundaries.

The key issue in local boundary search is how to find boundaries of constraints. This research builds on the concept of subset constraints boundary narrower (SCBN) function proposed by Bonyadi MR. (Bonyadi and Michalewicz 2014) The SCBN function was originally proposed to narrow a feasible region of COPs to its boundaries of constraints and particles within the narrowed band region can then be picked and stored during the optimization process of PSO. The area of band region is controlled by adjusting a positive parameter ε. The SCBN function is defined as (2):

$$H_{\Omega,\varepsilon} = \max\left\{\left|\max_{i\in\Omega}\{g_i(x)\}+\varepsilon\right|-\varepsilon, \max_{i\notin\Omega}\{g_i(x)\}\right\} \qquad (2)$$

From its formulation, $H_{\Omega,\ \varepsilon} \leq 0$ if and only if at least one of the constraints is 2ε-active in the subset Ω while others are satisfied. After using the SCBN function, particles moving near boundaries of constraints can be easily picked out and their positions will be provided for local boundary search.

Based on the SCBN function, this paper comes up with a parallel boundary search particle swarm optimization (PBSPSO) algorithm which has two parallel branches with cooperative mechanism. In one branch, constrained PSO based on the penalty function method is adopted and

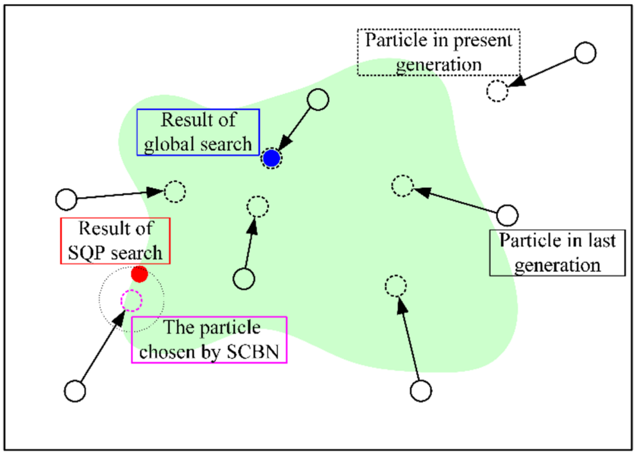
Particle in present generation
Result of global search
Result of SQP search
Particle in last generation
The particle chosen by SCBN

Fig. 1 Schematic diagram of PBSPSO algorithm. Green area represents feasible region, particles with black full lines are previous generation while others with dotted line are updated, the purple particle is the initial point of SQP search and it is chosen by SCBN function. Blue dot is the result of global search and red dot is the result of SQP search

global search is conducted. The particles near boundaries of constraints are picked out iteratively using the SCBN function and locations of these particles are stored. In another branch, the local boundary search process is performed from the stored positions as initial points based on the sequential quadratic programming (SQP). The results of local boundary search will reversely influence the search program of the other branch by guiding particles motion direction in the next generation. With the proposed algorithm, the cooperative mechanism of the global and local search processes not only directly increases the search intensity near boundaries of constraints, but also avoids the trial-and-error in adjusting penalty parameters. Note the increase of search intensity improves the chance of finding an optimal solution. The proposed algorithm is relatively suitable for real engineering optimization problems where solution often sits on the boundary of constraint.

This paper is organized as follows. Section 2 provides a brief introduction of the standard PSO algorithm. In Section 3, the proposed PBSPSO algorithm is described covering the constraint handling mechanism, diversity enhanced technique and the algorithm execution process. The result of numerical experiment is presented in Section 4. CEC06 benchmark test functions and five common engineering problems are chosen to test the proposed algorithm. In the last section, conclusions are drawn and the future research work is presented.

2 Standard particle swarm optimization

The basic particle swarm optimization algorithm was originally proposed by Kennedy and Eberhart (Kennedy and

Fig. 2 Flowchart of the proposed PBSPSO

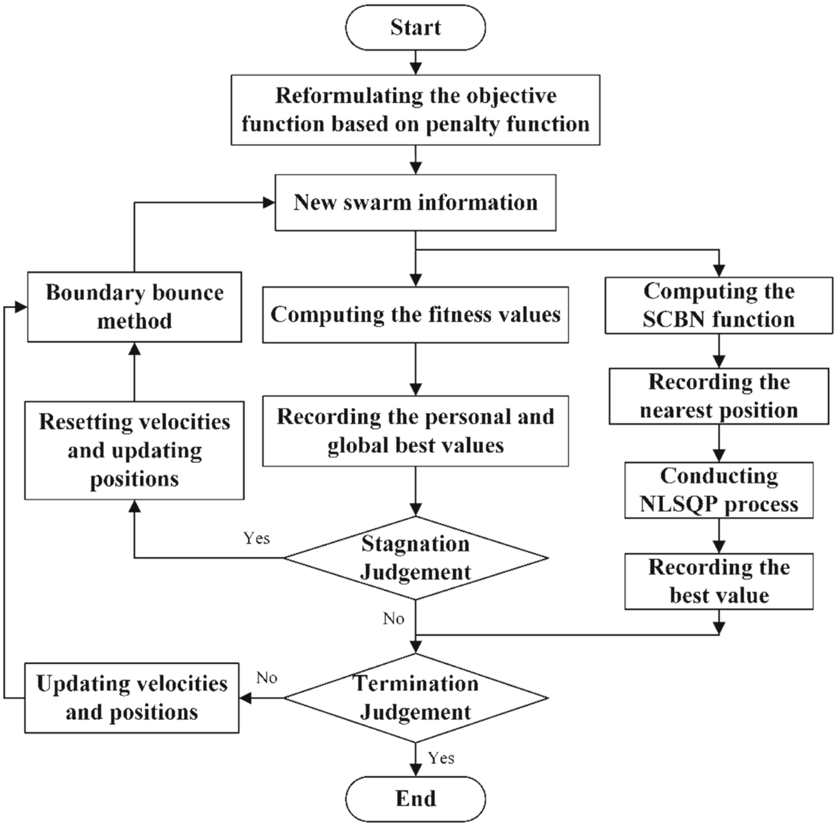
Start
Reformulating the objective function based on penalty function
New swarm information
Boundary bounce method
Computing the fitness values
Computing the SCBN function
Recording the nearest position
Resetting velocities and updating positions
Recording the personal and global best values
Conducting NLSQP process
Yes
Stagnation Judgement
Recording the best value
No
Updating velocities and positions
No
Termination Judgement
Yes
End

Table 1 Summary of 13 benchmark test problems in CEC06 (Liang 2006)

Problem	n	Type of function	ρ	LI	NI	LE	NE	a
g01	13	Quadratic	0.0111%	9	0	0	0	6
g02	20	Non-linear	99.9971%	0	2	0	0	1
g03	10	Polynomial	0.0000%	0	0	0	1	1
g04	5	Quadratic	52.1230%	0	6	0	0	2
g05	4	Cubic	0.0000%	2	0	0	3	3
g06	2	Cubic	0.0066%	0	2	0	0	2
g07	10	Quadratic	0.0003%	3	5	0	0	6
g08	2	Non-linear	0.8560%	0	2	0	0	0
g09	7	Polynomial	0.5121%	0	4	0	0	2
g10	8	Linear	0.0010%	3	3	0	0	6
g11	2	Quadratic	0.0000%	0	0	0	1	1
g12	3	Quadratic	4.7713%	0	1	0	0	0
g13	5	Non-linear	0.0000%	0	0	0	3	3

Notes of Table 1: n is the no. of dimensions, ρ is the percentage of feasible region in the whole search space, LI is the no. of linear inequality constraints, NI is the no. of nonlinear inequality constraints, LE is the no. of linear equality constraints, NE is the no. of nonlinear equality constraints, a is the no. of active constraints.

Eberhart 1995) in 1995 for unconstrained global optimization problems. After years of development, the standard PSO is widely used now and many modified PSO algorithms are revised from the standard version.

The mathematical expression of the standard PSO algorithm is summarized as follows. Assuming a problem with D-dimensions, x_k^i stands for the position of the i^{th} particle and v_k^i is the velocity. p_k^i and p_k^g are the historically best position of each particle and the global best position, respectively. The updating regulations of all particles are based on two rules: each particle will pursue the historical best position and global best position iteratively. The whole swarm is controlled by equations below. (3) is velocity update equation and (4) is position update equation.

$$v_{k+1}^i = \omega v_k^i + c_1 \cdot r_1 \cdot \left(p_k^i - x_k^i\right) + c_2 \cdot r_2 \cdot \left(p_k^g - x_k^i\right) \tag{3}$$

$$x_{k+1}^i = x_k^i + v_{k+1}^i \tag{4}$$

$$\omega(iter) = \frac{(iter_{\max} - iter)}{iter_{\max}} \times (\omega_{\max} - \omega_{\min}) + \omega_{\min} \tag{5}$$

There are three parts in (3). The first part contains the velocity of earlier generation and inertia factor ω. The latter ensures that particles are able to "fly" across the design space and also guarantees the balance between local and global search capacities. A linearly varying inertia weight proposed by Shi and Eberhart (Shi and Eberhart 1999) has significantly improved the standard PSO, shown as (5). *iter* is the current generation and $iter_{\max}$ is the maximum generation. The second part called the cognition component expresses the personality of each particle and motivates particles to pursue its personal best position in the search history. The third part is called the social component part (Kennedy 1997) and represents the collaborative behaviors among particles. c_1 and c_2 are named cognitive scaling parameter and social scaling parameter, respectively (Eberhart and Shi 2001). r_1 and r_2 are two random numbers within the range [0,1].

3 The parallel boundary search particle swarm optimization (PBSPSO) algorithm

Conventional constrained PSO algorithms ensure the feasibility of solutions by constraint handling mechanism, such as the penalty function method, but the process of adjusting parameters is complicated in various problems.

Table 2 Statistical performance of results of PBSPSO for 13 benchmark test functions

Funs	Best	Mean	Worst	SD	Nmax	Naver	Nmin	Pop
g01	−15	−15	−15	0.00E + 00	164	97	54	40
g02	−0.8036196	−0.7742606	−0.7271980	2.81E-02	737	413	300	100
g03	−1.0004754	−1.0001848	−0.9993826	2.37E-04	1293	472	124	100
g04	−30,665.539	−30,665.539	−30,665.539	7.43E-12	797	362	25	40
g05	5126.49671	5126.49671	5126.49671	1.86E-12	638	140	75	40
g06	−6961.814	−6961.814	−6961.814	4.96E-12	1709	803	149	40
g07	24.306	24.306	24.306	5.04E-14	111	91	79	40
g08	−0.095825	−0.095825	−0.095825	8.88E − 17	160	143	38	40
g09	680.630	680.630	680.630	4.64E − 13	992	738	539	40
g10	7049.248	7049.248	7049.248	2.04E − 12	1000	274	39	40
g11	0.7499	0.7499	0.7499	1.84E-13	12	12	12	40
g12	-1	-1	-1	0.00E + 00	95	78	62	40
g13	0.053941514	0.053941514	0.053941514	4.72E-15	1000	529	51	40

Table 3 Statistical performance of results of PBSPSO for the CEC06 benchmark test problems g01-g13

Fun.	Statistics	SMES (Mezura-Montes and Coello 2005)	HCOEA (Wang et al. 2007)	ATMES (Wang et al. 2008)	M-ABC (Mezura-Montes and Cetina-Dominguez 2012)	PSGA (Dhadwal et al. 2014)	MOPSO (Venter and Haftka 2010)	DSS-MDE (Zhang et al. 2008)	PBSPSO
g01	Best	−15.000	−15.000	−15.000	−15.000	−15.000	−14.98	−15.000	−15.000
	Mean	−15.000	−15.000	−15.000	−15.000	−15.000	−10.38	−15.000	−15.000
	Worst	−15.000	−14.999	−15.000	−15.000	−15.000	−6.00	−15.000	−15.000
	SD	0	4.30E-07	1.60E-14	0	0	2.54E + 00	0	0
g02	Best	−0.803601	−0.803241	−0.803388	−0.803615	−0.803597	−0.700	−0.803619	−0.8036196
	Mean	−0.785238	−0.801258	−0.790148	−0.799336	−0.794836	−0.539	−0.788011	−0.7742606
	Worst	−0.751322	−0.792363	−0.756986	−0.777438	−0.786442	−0.373	−0.744690	−0.7271980
	SD	1.67E-02	3.83E-03	1.30E-02	6.84E-03	5.64E-03	6.5E-02	1.50E-02	2.81E-02
g03	Best	−1.000	−1.000	−1.000	−1.000	−1.0005	–	−1.0005	−1.0004755
	Mean	−1.000	−1.000	−1.000	−1.000	−1.0005	–	−1.0005	−1.0001848
	Worst	−1.000	−1.000	−1.000	−1.000	−1.0005	–	−1.0005	−0.9993826
	SD	2.09E-04	1.30E-12	5.90E-05	4.68E-05	2.69E-05	–	−2.70E-09	2.37E-04
g04	Best	−30,665.539	−30,665.539	−30,665.539	−30,665.539	−30,665.539	−30,666	−30,665.539	−30,665.539
	Mean	−30,665.539	−30,665.539	−30,665.539	−30,665.539	−30,665.539	−30,664	−30,665.539	−30,665.539
	Worst	−30,665.539	−30,665.539	−30,665.539	−30,665.539	−30,665.539	−30,656	−30,665.539	−30,665.539
	SD	0	5.40E-07	7.40E-12	2.22E-11	7.28E-12	1.89E + 00	2.70E-11	7.43E-12
g05	Best	5126.599	5126.498	5126.498	5126.736	5126.497	–	5126.497	5126.497
	Mean	5174.492	5126.498	5127.648	5178.139	5140.897	–	5126.497	5126.497
	Worst	5304.167	5126.498	5135.256	5317.196	5166.438	–	5126.497	5126.497
	SD	50.06E + 00	1.73E-07	1.80E + 00	5.61E + 01	1.44E + 01	–	0	1.86E-12
g06	Best	−6961.814	−6961.814	−6961.814	−6961.814	−6961.814	−6959	−6961.814	−6961.814
	Mean	−6961.284	−6961.814	−6961.814	−6961.814	−6961.814	−6939	−6961.814	−6961.814
	Worst	−6952.482	−6961.814	−6961.814	−6961.814	−6961.814	−6910	−6961.814	−6961.814
	SD	1.85E + 00	8.51E-12	4.60E-12	0	9.26E-12	1.27E + 01	0	4.96E-12
g07	Best	24.327	24.306	24.306	24.315	24.360	26.97	24.306	24.306
	Mean	24.475	24.307	24.316	24.415	24.738	36.96	24.306	24.306
	Worst	24.843	24.309	24.359	24.854	24.999	72.54	24.306	24.306
	SD	1.32E-01	7.12E-04	1.10E-02	1.24E-01	2.30E-01	8.93E + 00	7.00E-08	5.04E-14
g08	Best	−0.095825	−0.095825	−0.095825	−0.095825	−0.095825	−0.096	−0.095825	−0.095825
	Mean	−0.095825	−0.095825	−0.095825	−0.095825	−0.095825	−0.096	−0.095825	−0.095825
	Worst	−0.095825	−0.095825	−0.095825	−0.095825	−0.095825	−0.095	−0.095825	−0.095825
	SD	0	2.42E-17	2.80E-17	4.23E-17	3.84E-09	0	3.90E-17	8.88E-17
g09	Best	680.632	680.630	680.630	680.632	680.631	681.8	680.630	680.630
	Mean	680.643	680.630	680.639	680.647	680.658	685.4	680.630	680.630
	Worst	680.719	680.630	680.673	680.691	680.725	693.5	680.630	680.630
	SD	1.55E-02	9.41E-08	1.00E-02	1.55E-02	2.48E-02	2.28E + 00	2.50E-13	4.64E-13
g10	Best	7051.903	7049.287	7052.253	7051.706	7049.255	7611	7049.248	7049.248
	Mean	7253.047	7049.525	7250.437	7233.882	7059.107	8992	7049.248	7049.248
	Worst	7638.366	7049.984	7560.224	7473.109	7092.609	15,553	7049.249	7049.248
	SD	1.36E + 02	1.50E-01	1.20E + 02	1.10E + 02	1.26E + 01	1.23E + 03	3.10E-04	2.04E-12
g11	Best	0.7500	0.7500	0.75	0.75	0.7499	–	0.7499	0.7499
	Mean	0.7500	0.7500	0.75	0.75	0.7499	–	0.7499	0.7499
	Worst	0.7500	0.7500	0.75	0.75	0.7499	–	0.7499	0.7499
	SD	1.52E-04	1.55E-12	3.40E-04	2.30E-05	1.92E-07	–	0	1.84E-13
g12	Best	−1.000	−1.000	−1.000	−1.000	−1.000	−1.000	−1.000	−1.000
	Mean	−1.000	−1.000	−1.000	−1.000	−1.000	−1.000	−1.000	−1.000
	Worst	−1.000	−1.000	−1.000	−1.000	−1.000	−1.000	−1.000	−1.000
	SD	0	0	1.00E-03	0	2.76E-09	0	0	0
g13	Best	0.053986	0.053949	0.053950	0.053985	0.054103	–	0.053942	0.05394151
	Mean	0.166385	0.053949	0.053959	0.158552	0.061601	–	0.053942	0.05394151
	Worst	0.468294	0.053949	0.053999	0.442905	0.083042	–	0.053942	0.05394151
	SD	1.77E-01	8.68E-08	1.30E-05	1.73E-01	7.23E-03	–	8.30E − 17	4.72E-15

Meanwhile, optimal solutions may be missed since the inadequate search near boundaries of constraints. To overcome these difficulties, in this work, the constrained PSO is modified based on a local boundary search procedure so that the parameter adjustment process is more straightforward and search capability near boundaries is observably improved. In this section, mechanism of the proposed parallel boundary search particle swarm optimization (PBSPSO) algorithm is introduced in detail.

The schematic diagram of the proposed PBSPSO algorithm is shown in Fig. 1. In one branch of PBSPSO, PSO algorithm based on non-stationary penalty function method (Yang et al. 1997) is adopted and PSO global search is conducted. In order to avoid the procedure from stagnating, a judgement criterion has been built and a velocity reset operator proposed in our previous study (Liu et al. 2016) is used to scatter particles away from the stagnation position. Boundary bounce method (Clerc 2006) is applied to pull back particles escaping from the given design space. In each generation, particles are calculated and picked by the SCBN function and location of the particle closest to boundaries of constraints is recorded for the other branch.

In the second branch, the sequential quadratic programming (SQP) is executed and the location information recorded in the first branch is set as the initial position of SQP local search. The Hessian matrix is calculated using the BFGS method for the SQP implementation. The results of SQP local search and PSO global search (red and blue dots in Fig. 1, respectively) are compared with each other and the better one is set to be the globally optimal solution of the present generation. When search result near boundary of constraint in the second branch is chosen as the current global optimal solution, the search direction of the PSO algorithm in the next generation will be influenced and particles in next generation will be led to the sensitive boundaries of constraint.

A cooperative mechanism is hereby established that the global search process of the PSO algorithm provides initial points iteratively for the local search process and the best solutions of local boundary search will influence the particles movement direction in return. The termination conditions of the proposed method are set as follows: the maximum generation is reached or relative difference between the results of last two generations is less than ε. The value ε is set as 10^{-5} in this study.

The parallel structure of PBSPSO algorithm has three major advantages than the existing algorithms. Firstly, local boundary search approach increases search intensity near boundaries of constraints so that there is no need to spend much time on adjusting the control parameter. Secondly, the local boundary search improves the overall search ability of the algorithm and increases the chances of finding a global optimal solution. Thirdly, two branches of the algorithm are not independent with each other while PSO provides initial location information for local boundary search and the results of local boundary search may lead PSO global search to sensitive boundaries of constraint. This treatment ensures the algorithm efficiency. Together, the parallel algorithm structure and the cooperative mechanism between global and local boundary search method effectively improved the optimization capability of the algorithm.

The algorithm flow chart is displayed in Fig. 2.

Table 4 The results of Wilcoxon test between PBSPSO and other algorithms

PBSPSO vs.	R+	R-	α = 0.05	α = 0.1	α = 0.2
SMES	80.5	10.5	H1	H1	H1
HCOEA	69.5	21.5	H0	H0	H1
ATMES	69.5	21.5	H0	H0	H1
M-ABC	80.5	10.5	H1	H1	H1
PSGA	62.5	28.5	H0	H0	H0
MOPSO	37.5	7.5	H0	H1	H1
DSS-MDE	50	41	H0	H0	H0

Notes of Table 2: The sample size of MOPSO is 9 while that of other algorithms is 13.

3.1 Constraint handling for the PBSPSO algorithm

As a constraint handling mechanism, the penalty function technique is simple in concept and convenient in operation. So, the widely-used non-stationary penalty functions (Yang et al. 1997) are employed in our work to handle constrain functions. The penalty functions are defined as in (6) and (7):

$$F(x) = f(x) + h(k)H(x), x \in S \tag{6}$$

$$H(x) = \sum_{i=1}^{m} \theta(q_i(x)) q_i(x)^{\gamma(q_i(x))} \tag{7}$$

Here $q_i(x) = \max\{0, g_i(x)\}$, $i = 1, \ldots, m$. $\theta(q_i(x))$ is the multi-stage assignment function. $\gamma(q_i(x))$ is the power of the penalty function and $g_i(x)$ represents the constrained functions. The rules of parameters are specified below (Eberhart and Shi 2001):

$$\begin{cases} \gamma(q_i(x)) = 1, \text{if } q_i(x) < 1 \\ \gamma(q_i(x)) = 2, \text{if } q_i(x) \geq 1 \end{cases}$$

$$\begin{cases} \theta(q_i(x)) = 10, \textit{if } q_i(x) < 0.001 \\ \theta(q_i(x)) = 20, \textit{if } q_i(x) \leq 0.1 \\ \theta(q_i(x)) = 100, \textit{if } q_i(x) \leq 1 \\ \theta(q_i(x)) = 300, \textit{if } q_i(x) > 1 \end{cases}$$

$h(k) = k\sqrt{k}$ for our optimization problem.

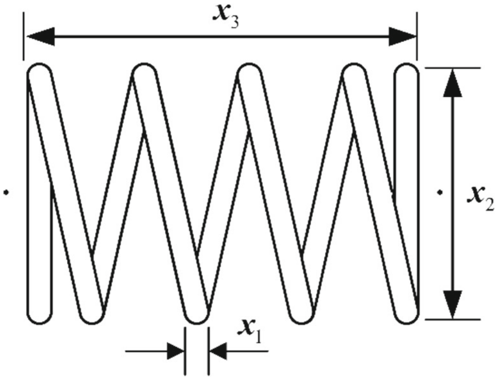

Fig. 3 Tension/Compression spring and its design variables

Table 5 Comparison of best solutions obtained from variable optimizers for tension/compression spring design problem

DV	DEDS (Zhang et al. 2008)	HEAA (Wang et al. 2009)	WCA (Eskandar et al. 2012a)	DELC (Wang and Li 2010)	G-QPSO (Coelho 2010)	MBA (Sadollah et al. 2013)	NM–PSO (Zahara and Kao 2009)	IAPSO (Guedria 2016)	PBSPSO
x1	0.051689	0.051690	0.051680	0.051689	0.051515	0.051656	0.051620	0.051685	0.051689
x2	0.356718	0.356729	0.356522	0.356718	0.352529	0.355940	0.355498	0.356629	0.356718
x3	11.288965	11.288294	11.300410	11.288966	11.538862	11.344665	11.333272	11.294175	11.288969
g1(x)	1.45E − 09	2.75E − 08	−1.65E − 13	−3.40E − 09	−4.83E − 05	0	1.01E − 03	−1.97E − 10	-1.29E-14
g2(x)	−1.20E − 09	−2.09E − 08	−7.9E − 14	2.45E − 09	−3.58E − 05	0	9.95E − 04	−4.64E − 10	6.66E-15
g3(x)	−4.053786	−4.053808	−4.053399	−4.053786	−4.0455	−4.052248	−4.061860	−4.053610	−4.053786
g4(x)	−0.727729	−0.727721	−0.727864	−0.727729	−0.73064	−0.728268	−0.728588	−1.091686	−0.727729
f(x)	0.01266523	0.01266523	0.012665	0.01266523	0.012665	0.012665	0.012630	0.01266523	0.01266523

3.2 Diversity enhancement in the PBSPSO algorithm

It has been noted that the PSO may be trapped into stagnation for numbers of generations without any improvement. For the purpose of escaping from stagnation, a velocity reset operator presented in our previous work (Liu et al. 2016) is employed.

Once the programming is judged to trap into stagnation, velocity reset operator is activated and velocity of swarm will be reset according to the current generation number and the inertia weight factor ω. The velocity reset equation is shown as (8).

$$\begin{aligned} V_{reset} &= \mu{\cdot}rw{\cdot}V_{rand} \\ \mu &= \frac{(iter_{\max}-iter_{current})}{iter_{\max}} \\ rw &= (rw_{\max}-rw_{\min})*\frac{(iter_{\max}-iter_{current})}{iter_{\max}}+rw_{\min} \end{aligned} \tag{8}$$

where V_{rand} is a velocity matrix in which elements are randomly defined between -$V_{\max}$ and $V_{\max}$.μ records the search process and will linearly decrease when the iterations increases. $iter_{\max}$ and $iter_{current}$ are the max generation and the current generation, respectively. rw is named velocity correlation coefficient and its boundary is within a predefined range $[rw_{\min}, rw_{\max}]$ referring to the inertia weight factor ω of the standard PSO. Along with the optimization process, $(iter_{\max} - iter_{current})$ decreases and the value of μ is diminished. rw is used to control the distribution range of reset velocities. Finally, V_{reset} shrinks and the algorithm's convergence property is guaranteed. Once the adaptive reset operator is activated, velocities are reset by (8) and particles are rebounded off the stagnation position by (9).

$$P_p = P_{stagnation} + V_{reset} \tag{9}$$

Table 6 Comparison of Statistical performance of results given by different optimizers for tension/compression spring design problem

Method	Worst	Mean	Best	S.D
GA2 (Coello and Montes 2002)	0.012973	0.012742	0.012681	5.90E − 05
CAEP (Coello and Becerra 2004)	0.015116	0.013568	0.012721	8.42E − 04
CPSO (He and Wang 2007a)	0.012924	0.012730	0.012675	5.20E − 04
HPSO (He and Wang 2007b)	0.012719	0.012707	0.012665	1.58E − 05
NM–PSO (Zahara and Kao 2009)	0.012633	0.012631	0.012630	8.47E − 07
G-QPSO (Coelho 2010)	0.017759	0.013524	0.012665	1.27E-03
DE (Lampinen 2002)	0.012790	0.012703	0.012670	2.7E − 05
DELC (Wang and Li 2010)	0.012665	0.012665	0.012665	1.3E − 07
DEDS (Zhang et al. 2008)	0.012738	0.012669	0.012665	1.3E − 05
HEAA (Wang et al. 2009)	0.012665	0.012665	0.012665	1.4E − 09
PSO–DE (Liu et al. 2010)	0.012665	0.012665	0.012665	1.2E − 08
LCA (Kashan 2011)	0.01266667	0.01266541	0.01266523	3.88E − 07
WCA (Eskandar et al. 2012a)	0.012952	0.012746	0.012665	8.06E − 05
MBA (Sadollah et al. 2013)	0.012900	0.012713	0.012665	6.30E − 05
APSO (Yang 2010)	0.014937	0.013297	0.012700	6.85e − 4
IAPSO (Guedria 2016)	0.01782864	0.013676527	0.01266523	1.573E − 3
ABC (Cuevas and Cienfuegos 2014)	0.01321041	0.012850718	0.01266523	1.18E-04
PBSPSO	0.01266523	0.01266523	0.01266523	5.10E-17

Table 7 Comparison of best solutions obtained from variable optimizers for three bar truss design problem

DV	DEDS (Zhang et al. 2008)	PSO-DE (Liu et al. 2010)	WCA (Eskandar et al. 2012a)	MBA (Sadollah et al. 2013)	SCA (Ray and Liew 2003)	PBSPSO
x1	0.788675	0.788675	0.788651	0.788565	0.788621037	0.788675137
x2	0.408248	0.408248	0.408316	0.4085597	0.408401334	0.408248283
g1(x)	−2.10E-11	−5.75E-13	4.23E-07	−5.29E-11	−8.28E-09	−6.66E-16
g2(x)	−1.464102	−1.464102	−1.464024	−1.4637475	−1.46392765	−1.464101624
g3(x)	−0.535898	−0.535898	−0.535975	−0.5362524	−0.536072358	−0.535898376
f(x)	263.895843	263.895843	263.895843	263.8958522	263.8958466	263.895843

3.3 Implementation of the PBSPSO algorithm

The implementation of the proposed PBSPSO algorithm is described as follows.

```
1:   Reformulating the objective function according to Eq. (6) and Eq. (7)
2:   Initializing the swarm position
3:   Obtaining new swarm information

The branch of constrained PSO algorithm:
4:   Computing the fitness values
5:   Recording the personal and global best values
6:   Making stagnation judgement
7:   If procedure stagnates
8:       Resetting Velocities and updating positions
9:       according to Eq. (8) and Eq. (9)
10:      Conducting boundary bounce method
11:      Turning back to Step 3
12:  Else
13:      Putting the best solution out as Output_1

The branch of local boundary search:
4':  Calculating particles according to SCBN
5':  function expressed in Eq. (2)
6':  Recording the particle nearest to boundary
7':  Conducting SQP process at the selected particle
8':  Comparing the present result with the historical
9':  best result of SQP
10': Putting the best solution out as Output_2

14:  Comparing Output_1 with Output_2
15:  Conducting termination judgement
16:  If Termination condition is not satisfied
17:      Updating particles velocities and positions according to Eq. (3), Eq. (4) and Eq. (5)
18:      Conducting boundary bounce method
19:      Turning back to Step 3
20:  Else
21:      Putting the present global best values as the optimum solution
```

4 Numerical experiment

In this section, thirteen benchmark test functions from CEC06 (Liang 2006) and five real world engineering problems were chosen to test the proposed algorithm. The benchmark test functions from CEC06 are used to test the ability of the proposed algorithm to deal with conventional constrained optimization problems while five engineering problems are used to evaluate its ability to solve practical engineering optimization problems.

Table 8 Comparison of Statistical performance of results given by different optimizers for tension/compression spring design problem

Method	Worst	Mean	Best	S.D
PSO-DE (Liu et al. 2010)	263.895843	263.895843	263.895843	4.50E-10
DEDS (Zhang et al. 2008)	263.895849	263.895843	263.895843	9.70E-07
HEAA (Wang et al. 2009)	263.896099	263.895865	263.895843	4.90E-05
WCA (Eskandar et al. 2012a)	263.896201	263.895903	263.895843	8.71E-05
MBA (Sadollah et al. 2013)	263.915983	263.897996	263.895852	3.93E-03
SCA (Ray and Liew 2003)	263.969756	263.903357	263.895847	1.30E-02
PBSPSO	263.895843	263.895843	263.895843	5.80E-14

The initial population is set to be 40 for both test functions and engineering problems. The population of g02 and g03 is 100 because of the strong nonlinearity of these problems. The thickness parameter ε in SCBN is set as 0.5 and tolerance δ for the translation of equality constraints into inequalities is set as 1.0E-4. 25 runs have been completed for each problem. The tests were run on a computer with 3.40GHz core i7 CPU and 8 GB RAM.

4.1 Benchmark test functions

The chosen 13 functions from CEC06 are characterized as several types based on the number of variables, degree of nonlinearity, proportion of feasible region, constraint type and the number of active constraints. See Table 1.

Table 2 shows the results of benchmark test functions of PBSPSO and contains the best, mean, worst, SD values, the maximum, average and minimum iteration numbers and the initial population size for different functions. It must be noted that the average iteration of four functions are below 100 while that of ten functions are below 500, which indicates relatively high levels of efficiency of the proposed algorithm.

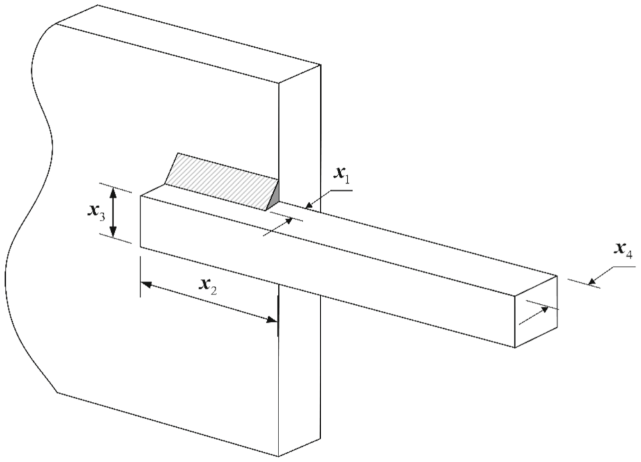

Fig. 4 Welded beam and its design variables

The proposed algorithm is compared with the existing most advanced algorithms including SMES (Mezura-Montes and Coello 2005), HCOEA (Wang et al. 2007), ATMES (Wang et al. 2008), M-ABC (Mezura-Montes and Cetina-Dominguez 2012), PSGA (Dhadwal et al. 2014), MOPSO (Venter and Haftka 2010) and DSS-MDE (Zhang et al. 2008). Instead of using the conventional penalty function, SMES (Mezura-Montes and Coello 2005) is based on a diversity mechanism and feasibility-based comparison mechanism to handle infeasible solutions and guide particles to the feasible region. HCOEA (Wang et al. 2007) effectively combines multi-objective optimization with a niching genetic algorithm for global search and a parallel local search adopting a clustering partition of the population and multiparent crossover. To achieve a balance between optimizing the objective function and reducing constraint violations, an adaptive tradeoff model (ATM) is combined with the evolutionary strategy (ES) in ATMES (Wang et al. 2008). M-ABC (Mezura-Montes and Cetina-Dominguez 2012) is a constrained optimization algorithm containing four modifications such as selection mechanism, the scout bee operator and the equality constraint and constraints of boundaries. PSGA (Dhadwal et al. 2014) is a novel hybrid evolutionary algorithm composed of PSO and GAs, in which PSO is aimed for improving the worst solutions and efficiency while GAs is to keep population diversity. A constrained, single objective optimization problem is converted into an unconstrained, bi-objective problem in Multi-objective Particle Swarm Optimization (Venter and Haftka 2010) (MOPSO) algorithm and the method does not need any problem dependent parameters while it provides performance that is similar or superior to that of a tuned penalty function approach. Considering the promise in feasible solutions during the evolution process, DSS-MDE (Zhang et al. 2008) assembles the dynamic stochastic selection (DSS) with multimember differential evolution.

The comparison is made based on the statistical performance of each algorithm. The PBSPSO achieves the present optimum solutions for all test functions except for g03. Apart from the optimum, mean and worst values of PBSPSO are also better than using other algorithms in full. The results of

Table 9 Comparison of best solutions obtained from variable optimizers for welded beam design optimization problem

DV	GA2 (Coello and Montes 2002)	CPSO (He and Wang 2007a)	CAEP (Coello and Becerra 2004)	NM-PSO (Zahara and Kao 2009)	WCA (Eskandar et al. 2012a)	MBA (Sadollah et al. 2013)	IAPSO (Guedria 2016)	PBSPSO
x1	0.205986	0.202369	0.205700	0.205830	0.205728	0.205729	0.2057296	0.2057296
x2	3.471328	3.544214	3.470500	3.468338	3.470522	3.470493	3.47048866	3.4704887
x3	9.020224	9.048210	9.036600	9.036624	9.036620	9.036626	9.03662391	9.0366239
x4	0.206480	0.205723	0.205700	0.20573	0.205729	0.205729	0.20572964	0.2057296
g1(x)	−0.074092	−12.839796	−0.000472	−0.02525	−0.034128	−0.001614	−1.05E − 10	−4.00E-11
g2(x)	−0.266227	−1.247467	−0.001561	−0.053122	−3.49E − 05	−0.016911	−6.91E − 10	−1.20E-10
g3(x)	−0.000495	−0.001498	0	0.000100	−1.19e − 06	−2.40E − 07	−7.66E − 15	0
g4(x)	−3.430043	−3.429347	−3.433213	−3.433169	−3.432980	−3.432982	−3.43298378	−3.4329838
g5(x)	−0.080986	−0.079381	−0.080700	−0.080830	−0.080728	−0.080729	−0.08072964	−0.0807296
g6(x)	−0.235514	−0.235536	−0.235538	−0.235540	−0.235540	−0.235540	−0.23554032	−0.23554032
g7(x)	−58.666440	−11.681355	2.603347	−0.031556	−0.013503	−0.001464	−5.80E − 10	−5.00E-11
f(x)	1.728226	1.728024	1.724852	1.724717	1.724856	1.724853	1.7248523	1.7248523

g02 and g03 are not good because the two problems are of strong nonlinearity. Overall, PBSPSO is strong to be competitive against other algorithms on CEC06 benchmark test functions. The results are displayed in Table 3 and numbers in bold represent optimal results.

To detect significant differences between algorithms in Table 3, the Wilcoxon matched-pairs signed-ranks test (Garcia et al. 2009), a non-parametric pairwise statistical test has been made in this study. The Wilcoxon tests have been conducted with three levels of significance including $\alpha = 0.05$, $\alpha = 0.1$ and $\alpha = 0.2$. The results of Wilcoxon test are listed in Table 4. The statistical hypotheses for the Wilcoxon test are as follows:

(1) The null hypothesis H_0: The results of two algorithms come from one statistical population.

(2) The alternative hypothesis H_1: The results of two algorithms come from different statistical populations.

According to results of Wilcoxon test in Table 4, conclusions can be drawn:

(1) The PBSPSO algorithm outperforms SMES and M-ABC(2/7) with a level of significance $\alpha = 0.05$.

(2) The PBSPSO algorithm outperforms SMES, M-ABC and MOPSO(3/7) with a level of significance $\alpha = 0.1$.

(3) The PBSPSO algorithm outperforms SMES, M-ABC, MOPSO, HCOEA and ATMES(5/7) with a level of significance $\alpha = 0.2$.

Table 10 Comparison of Statistical performance of results given by different optimizers for welded beam design optimization problem

Method	Worst	Mean	Best	SD
GA2 (Coello and Montes 2002)	1.993408	1.792654	1.728226	7.47E − 02
CAEP (Coello and Becerra 2004)	3.179709	1.971809	1.724852	4.43E − 01
CPSO (He and Wang 2007a)	1.782143	1.748831	1.728024	1.29E − 02
HPSO (He and Wang 2007c)	1.814295	1.749040	1.724852	4.01E − 02
PSO-DE (Liu et al. 2010)	1.724852	1.724852	1.724852	6.70E − 16
NM-PSO (Zahara and Kao 2009)	1.733393	1.726373	1.724717	3.50E − 03
WCA (Eskandar et al. 2012a)	1.744697	1.726427	1.724856	4.29E − 03
LCA (Kashan 2011)	1.7248523	1.7248523	1.7248523	7.11E − 15
MBA (Sadollah et al. 2013)	1.724853	1.724853	1.724853	6.94E − 19
APSO (Yang 2010)	1.993999	1.877851	1.736193	0.076118
IAPSO (Guedria 2016)	1.7248624	1.7248528	1.7248523	2.02E − 06
SSO-C (Cuevas and Cienfuegos 2014)	1.7993318	1.7464616	1.7248523	2.57E-02
FA (Gandomi et al. 2011)	2.3455793	1.8786560	1.7312065	2.68E-01
PBSPSO	1.7248523	1.7248523	1.7248523	4.10432E-15

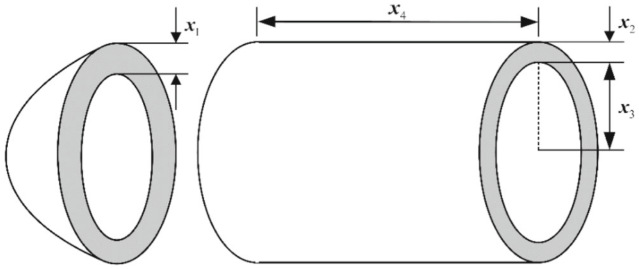

Fig. 5 Pressure vessel and its design variables

By means of the Wilcoxon test, the statement: "PBSPSO is strong to be competitive against other algorithms." could be statistically supported.

4.2 Engineering problems

Five common engineering problems have been tested in this part.

4.2.1 Tension/compression spring design optimization problem

Tension/compression spring design optimization problem aims at minimizing the weight of the spring constrained on minimum deflection, shear stress, surge frequency and outside diameter. Three variables are the wire diameter (x_1), the mean coil diameter (x_2) and the number of active coils (x_3), respectively. Fig. 3 presents a schematic view of the problem. The mathematical model of the problem is shown in Appendix 1.

The proposed PBSPSO is compared with a number of advanced constrained optimization algorithms including DEDS (Zhang et al. 2008), HEAA (Wang et al. 2009), WCA (Eskandar et al. 2012a), DELC (Wang and Li 2010), G-QPSO (Coelho 2010), MBA (Sadollah et al. 2013), NM-PSO (Zahara and Kao 2009), IAPSO (Guedria 2016), GA2 (Coello and Montes 2002), CAEP (Coello and Becerra 2004), CPSO (He and Wang 2007a), HPSO (He and Wang 2007b), DE (Lampinen 2002), PSO-DE (Liu et al. 2010), LCA (Kashan 2011), APSO (Yang 2010), ABC (Cuevas and Cienfuegos 2014). The best solutions and the statistical performance of results are summarized in Table 5 and Table 6, respectively.

The active constraints of the problem are g1 and g2. The values of the two constraints in best results of WCA, MBA and PBSPSO are closer to zero than others, which indicates that the optimal solutions of these three algorithms impend over boundaries of active constraints. In other words, deflection and shear stress are quite close to the minimum deflection (represented by g1) and the limited shear stress (represented by g2), respectively. The mean and the worst results of PBSPSO are also better than most of other algorithms. The standard deviation results show that results centrality of PBSPSO is greater. All the 25 runs finished within 100 iterations.

4.2.2 Three bar truss design problem

The problem aims at minimizing the total volume of bar truss system. Refer to Appendix 2 for mathematical form of the problem.

The best solutions and statistic results of PBSPSO are compared with DEDS (Zhang et al. 2008), PSO-DE (Liu et al. 2010), WCA (Eskandar et al. 2012a), MBA (Sadollah et al. 2013), SCA (Ray and Liew 2003), HEAA (Wang et al. 2009), see Table 7 and Table 8.

The active constraint of the problem is g1. The best solutions of DEDS, PSO-DE and PBSPSO are closer to boundaries of constraints and are also better than other algorithms. The mean, worst and SD values of PBSPSO are better than all of others. It should be noted that the SD value of PBSPSO is 1E-14 order of magnitude, which is much smaller and far better than other algorithms. The maximum iteration number is 51 in 25 runs.

Table 11 Comparison of best solutions obtained from variable optimizers for pressure vessel design optimization problem

DV	CDE (Fz et al. 2007)	GA2 (Coello and Montes 2002)	NM-PSO (Zahara and Kao 2009)	G-QPSO (Coelho 2010)	WCA (Eskandar et al. 2012a)	MBA (Sadollah et al. 2013)	APSO (Yang 2010)	IAPSO (Guedria 2016)	PBSPSO
x1	0.8125	0.8125	0.8036	0.8125	0.7781	0.7802	0.8125	0.8125	0.77817
x2	0.4375	0.4375	0.3972	0.4375	0.3846	0.3856	0.4375	0.4375	0.38465
x3	42.0984	42.0974	41.6392	42.0984	40.3196	40.4292	42.0984	42.0984	40.31962
x4	176.6377	176.6540	182.4120	176.6372	200.0000	198.4964	176.6374	176.6366	200.00000
g1(x)	−6.67E − 07	−2.01E − 03	3.65E − 05	−8.79E − 07	−2.95E − 11	0	−9.54E − 7	−4.09E − 13	0
g2(x)	−3.59E − 02	−3.58E − 02	3.79E − 05	−3.58E − 02	−7.15E − 11	0	−3.59E − 2	−3.58E − 2	−3.89E-16
g3(x)	−3.683016	−27.8861	−1.5914	−0.2179	−1.35E − 06	−86.3645	−63.3626	−1.39E − 07	1.16E-09
g4(x)	−63.3623	−63.3460	−57.5879	−63.3628	−40.0000	−41.5035	−0.9111	−63.3634	−40.00000
f(x)	6059.7340	6059.9463	5930.3137	6059.7208	5885.3327	5889.3216	6059.72418	6059.71433	5885.33277

Table 12 Comparison of Statistical performance of results given by different optimizers for pressure vessel design optimization problem

Method	Worst	Mean	Best	SD
GA2 (Coello and Montes 2002)	6469.3220	6177.2533	6059.9463	130.9297
NM-PSO (Zahara and Kao 2009)	5960.0557	5946.7901	5930.3137	9.1610
G-QPSO (Coelho 2010)	7544.4925	6440.3786	6059.7208	448.4711
CDE (Fz et al. 2007)	6371.0455	6085.2303	6059.7340	43.0130
WCA (Eskandar et al. 2012a)	6590.2129	6198.6172	5885.3327	213.0490
LCA (Kashan 2011)	6090.6114	6070.5884	6059.8553	11.37534
MBA (Sadollah et al. 2013)	6392.5062	6200.64765	5889.3216	160.34
APSO (Yang 2010)	7544.49272	6470.71568	6059.7242	326.9688
IAPSO (Guedria 2016)	6090.5314	6068.7539	6059.7143	14.0057
PBSPSO	5885.3328	5885.3328	5885.3328	9.28E-13

4.2.3 Welded beam design optimization problem

The welded beam design optimization problem aims to minimize the fabrication cost of welded beam. The problem is constrained by shear stress, bending stress in the beam, buckling load on the bar and deflection on the beam. The design variables include the thickness of the welding line (x_1), length of the weld (x_2), beam thickness (x_3) and width of the beam (x_4). Fig. 4 shows the schematic view of the problem. Mathematical description of the problem is presented in Appendix 3.

Algorithms chosen for comparison in this problem include: GA2 (Coello and Montes 2002), CPSO (He and Wang 2007a), CAEP (Coello and Becerra 2004), NM-PSO (Zahara and Kao 2009), WCA (Eskandar et al. 2012b), MBA (Sadollah et al. 2013), IAPSO (Guedria 2016), HPSO (He and Wang 2007b), PSO-DE (Liu et al. 2010), LCA (Kashan 2011), APSO (Yang 2010),SSO-C (Cuevas and Cienfuegos 2014) and FA (Gandomi et al. 2011). See Table 9 and Table 10.

The active constraints of the problem are g1, g2 and g3. The optimal solutions of IAPSO and PBSPSO are closer to the boundaries of constraints defined by g1, g2 and g3 than any other solutions and the optimal solutions of the two algorithms are also the best. The result of CAEP is superior but not feasible. Statistic results show that the optimal solution of PBSPSO algorithm is superior to 13 state-of-the-art algorithms in Table 10. The maximum iteration number is 81 in 25 runs among all algorithms.

4.2.4 Pressure vessel design optimization problem

Pressure vessel design optimization problem is considered here as the fourth problem. The target of the problem is to minimize the total fabricating cost of the pressure vessel. The constraints include materials, forming and welding costs. The design variables are thickness of the shell (x_1), thickness of the head (x_2), the inner radius (x_3) and the length of the cylindrical section of the vessel (x_4) as plotted in Fig. 5. Mathematical description is presented in Appendix 4.

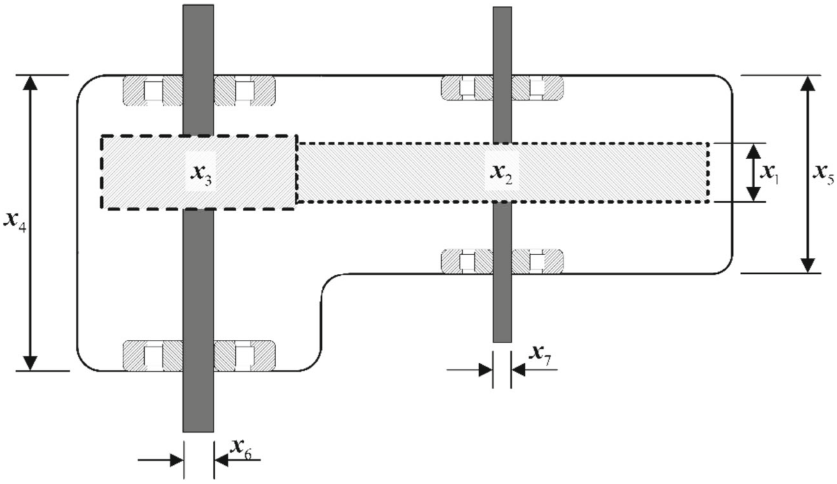

Fig. 6 Speed reducer and its design variables

Table 13 Comparison of best solutions obtained from variable optimizers for speed reducer design optimization problem

DV	DEDS (Zhang et al. 2008)	HEAA (Wang et al. 2009)	PSO-DE (Liu et al. 2010)	WCA (Eskandar et al. 2012a)	MBA (Sadollah et al. 2013)	IAPSO (Guedria 2016)	SCA (Ray and Liew 2003)	ABC (Akay and Karaboga 2012)	PBSPSO
x1	3.5	3.500022	3.5	3.5	3.5	3.5	3.5	3.5	3.5
x2	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7
x3	17	17	17	17	17	17	17	17	17.0
x4	7.3	7.300428	7.3	7.3	7.300033	7.3	7.32760205	7.3	7.3
x5	7.715320	7.715377	7.800000	7.715319	7.715772	7.7153199	7.71532175	7.8	7.7153199
x6	3.350215	3.350231	3.350214	3.350214	3.350218	3.350215	3.35026702	3.350215	3.3502147
x7	5.286654	5.286664	5.2866832	5.286654	5.286654	5.286654	5.28665450	5.286683	5.2866545
g1(x)	−0.073915	−0.073923	−0.073915	−0.073915	−0.073915	−0.073915	−0.073917	−0.073915	−0.07391528
g2(x)	−0.197999	−0.198006	−0.197999	−0.197999	−0.197999	−0.197999	−0.198	−0.197999	−0.19799853
g3(x)	−0.499172	−0.499095	−0.499172	−0.499172	−0.499167	−0.499172	−0.493501	−0.499172	−0.49917225
g4(x)	−0.904644	−0.904643	−0.901472	−0.904644	−0.904627	−0.904644	−0.904644	−0.901555	−0.9046439
g5(x)	−2.99E-07	−1.40E-05	−4.73E-13	5.96E-07	−2.93E-06	4.01E-13	−6.36E-07	−2.99E-07	−2.00E-15
g6(x)	2.64E-07	−5.00E-06	−4.67E − 14	2.64E-07	3.51E-07	−1.51E-12	-1.95E-08	−0.000633	1.11E-15
g7(x)	−0.7025	−0.7025	−0.7025	−0.7025	−0.7025	−0.7025	−0.7025	−0.7025	−0.7025
g8(x)	0	−6.00E-06	0	0	0	0	−1.93E-06	0	0
g9(x)	−0.583333	−5.83E-01	−0.583333	−0.583333	−0.583333	−0.583333	−0.583333	−0.583333	−0.58333333
g10(x)	−0.051326	−5.14E-02	−0.051326	−0.051326	−0.051329	−0.051326	−0.054889	−0.051326	−0.05132575
g11(x)	−7.78E-08	−6.00E-06	−0.010852	5.18E-08	−5.87E-05	1.49E-09	−2.33E-07	−0.010695	−8.88E-16
f(*x*)	2994.47107	2994.49911	2996.34817	2994.47107	2994.48245	2994.47107	2994.7442	2997.058412	2994.34132

The optimum and statistical performance of results of PBSPSO are compared with CDE (Fz et al. 2007), GA2 (Coello and Montes 2002), G-OPSO (Coelho 2010),WCA (Eskandar et al. 2012b), MBA (Sadollah et al. 2013), APSO (Yang 2010), IAPSO (Guedria 2016), NM-PSO (Zahara and Kao 2009), G-QPSO (Coelho 2010) and LCA (Kashan 2011) as summarized in Table 11 and Table 12.

The active constraints are g1, g2 and g3 in this problem as shown in Table 11. The best solutions of WCA and PBSPSO are closer to the three boundaries of constraints and the performance of the two algorithms precedes others for this problem. The best solution of MBA is quite close to two of the active constraints (g1 and g2), however, it is far from the boundary of constraint defined by g3. As a result, it is worse than solutions of WCA and PBSPSO. The statistic results shown in Table 12 demonstrate that the solution of PBSPSO possesses a strong centrality and is superior to other constrained algorithms for this problem. The maximum iteration number is 69 in 25 runs.

4.2.5 Speed reducer design optimization problem

This problem aims to minimize the mass of the speed reducer as shown in Fig. 6. The problem is constrained by the bending stress of gear teeth, the surface stress, transverse deflections of the shafts and stress in the shafts. Variables are the 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 diameter of the first (x_6) and the second shafts (x_7), respectively. See Appendix 5 for mathematical expression.

Table 14 Comparison of Statistical performance of results given by different optimizers for speed reducer design optimization problem

Method	Worst	Mean	Best	SD
SC (Ray and Liew 2003)	3009.964736	3001.758264	2994.744241	4.0
PSO-DE (Liu et al. 2010)	2996.348204	2996.348174	2996.348167	6.4E − 06
DEDS (Zhang et al. 2008)	2994.471066	2994.471066	2994.471066	3.6E − 12
HEAA (Wang et al. 2009)	2994.752311	2994.613368	2994.499107	7.0E − 02
ABC (Akay and Karaboga 2012)	NA	2997.058000	2997.058000	0.0
WCA (Eskandar et al. 2012a)	2994.505578	2994.474392	2994.471066	7.4E − 03
LCA (Kashan 2011)	2994.471066	2994.471066	2994.471066	2.66E − 12
MBA (Sadollah et al. 2013)	2999.652444	2996.769019	2994.482453	1.56
APSO (Yang 2010)	4443.017639	3822.640624	3187.630486	366.146
IAPSO (Guedria 2016)	2994.471066	2994.471066	2994.471066	2.65E − 09
PBSPSO	2994.3413157	2994.3413157	2994.3413157	4.64E-13

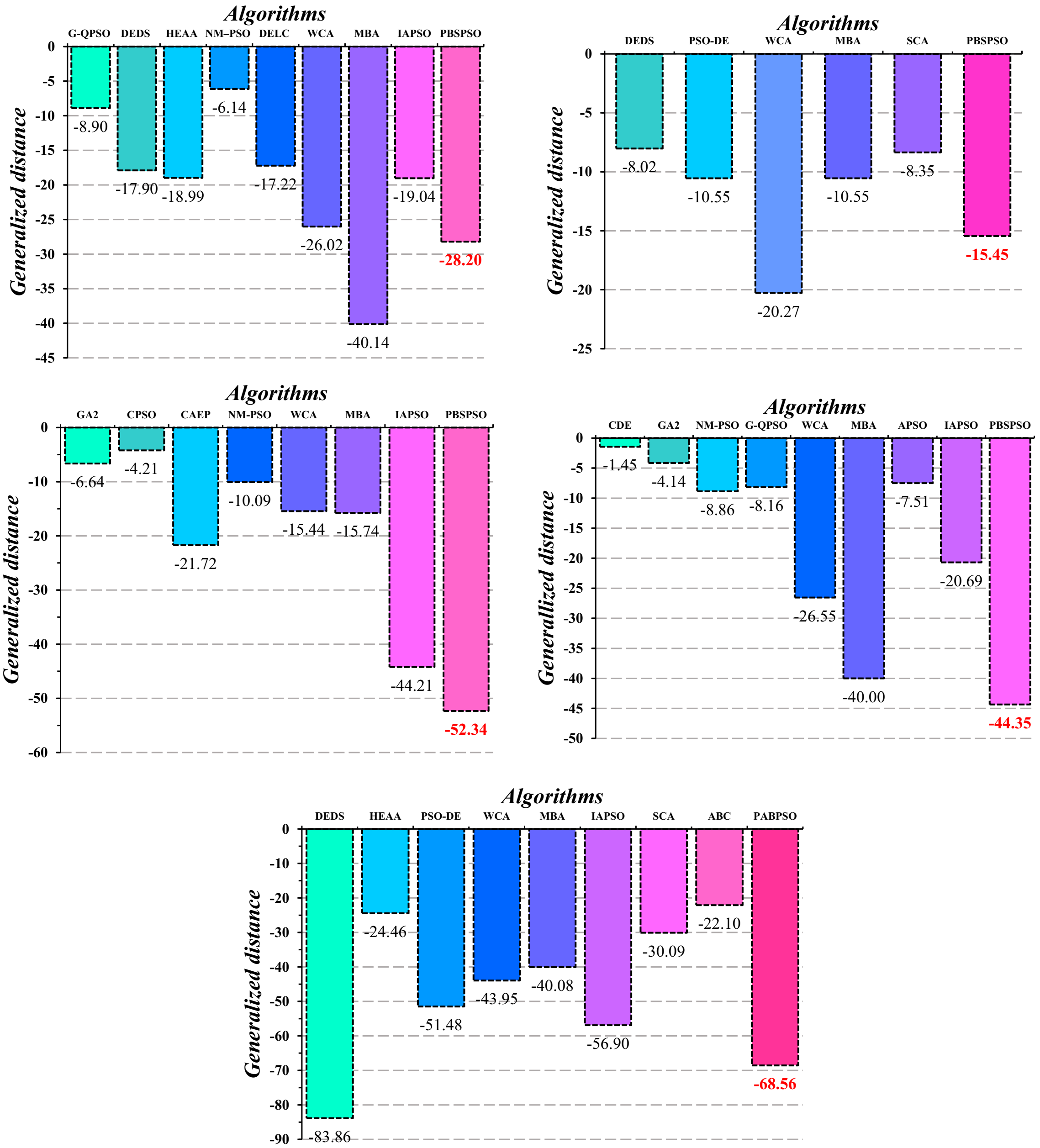

Fig. 7 Generalized distance of five engineering problems

The following algorithms are involved in comparison of this problem: DEDS (Zhang et al. 2008), HEAA (Wang et al. 2009), PSO-DE (Liu et al. 2010), WCA (Eskandar et al. 2012b), MBA (Sadollah et al. 2013), LCA (Kashan 2011), IAPSO (Guedria 2016), SCA (Ray and Liew 2003), ABC (Akay and Karaboga 2012), SC (Ray and Liew 2003) and APSO (Yang 2010). See Table 13 and Table 14.

There are 4 active constraints including g5, g6, g8 and g11. The best solution of PBSPSO is closer to the boundaries of active constraints as indicated by values of active constraints. The best solution of ABC is the worst in all algorithms because the solution is respectively far from the boundary of the active constraint (g11) indicating that resource expressed by g11 was not made full use of. The mean and worst solutions of PBSPSO are also better than other algorithms and the S.D value is relatively small. The maximum iteration number is 45 for this problem.

4.2.6 Further discussion

As mentioned before, optimal solutions of actual engineering problems generally locate on boundaries of active constraints because of the limitation in resources (Bonyadi and Michalewicz 2014). In some ways, the close distance in which solutions get to boundaries of constraints indicates the strong capability algorithms possess to conduct boundary search. As a result, solutions found are relatively superior. In order to quantitatively describe this significant characteristic of constrained optimization algorithms, a generalized distance $\tilde{\mathrm{D}}$ is defined in this article as (10).

$$\tilde{\mathrm{D}} = \sum_i D_i \tag{10}$$

$\mathrm{D}_i$is formulated as (11).

$$\mathrm{D}_i = \begin{cases} 0 \quad , if\,|g_i(x)|\geq 1 \\ \log_{10}|g_i(x)|, if\,10^{-20} < |g_i(x)| < 1 \\ -20 \quad , if\,|g_i(x)|\leq 10^{-20} \end{cases} \tag{11}$$

i is the number of constraints. $|g_i(x)|$ is the absolute value of the ith constraint and there are three components in (11). If $|g_i(x)| \geq 1$, it means the solution is far away from the ith constraint, D_i equals zero. If $10^{-20} < |g_i(x)| < 1$, the solution is near the ith constraint and D_i is defined as $\log_{10}|g_i(x)|$. If $f|g_i(x)| \leq 10^{-20}$, the solution is almost on the ith boundary of constraint and D_i is set to be −20. The value of $\tilde{\mathrm{D}}$ is a negative real number without unit. The smaller the value, the closer the solutions get to boundaries of constraints. The $\tilde{\mathrm{D}}$ values in the above 5 engineering problems are presented using the bar charts as shown in Fig. 7.

As displayed in Fig. 7, generalized distance values of PBSPSO are relatively small in all the five problems, which indicates that the optimal solutions are close to active constraints and the algorithm possesses strong capability of boundary search. Summarized the overall performance of the modified constrained PSO algorithms in boundary search, the proposed algorithm behaves better than other modified PSO algorithms such as G-QPSO, IAPSO, CPSO and APSO. Although some algorithms, e.g., MBA, WCA and DEDS are better than PBSPSO in some problems, their performance fluctuates from one problem to another, which reveals the relative instability of these algorithms in different engineering problems. For example, MBA performs well in the spring design problem and the vessel design problem. In other three problems, its solutions are ordinary while performance of PBSPSO is stable and competitive in all problems.

5 Conclusions and future research work

In real world engineering optimization problems, Optimal solutions generally locate on boundaries of active constraints because of limitation of resources. Considering this characteristic, a parallel boundary search particle swarm optimization (PBSPSO) method is proposed as a novel constrained optimization algorithm in this paper. The PBSPSO has two branches with cooperative mechanism. In the global search branch, PSO with the penalty function method conducts the search process. The velocity reset operator in our prior development of PSO algorithm is adopted to enhance swarm diversity. Particles are calculated by the SCBN function and locations of those close to the boundaries of constraints are selected as initial points in the branch of local boundary search. In the local boundary search branch, SQP proceeds at the points identified by the former. The results from the local boundary search procedure will reversely influence the global search process in the next generation and particles will be led to boundaries of active constraints. The proposed algorithm relieves the complex parameters adjustment course in traditional constrained PSO methods. Its cooperative mechanism increases the chance of finding global optimal solutions, which is suitable for practical engineering constrained optimization problems. PBSPSO is tested by thirteen CEC06 benchmark test functions and five common engineering optimization problems. Conclusions are summarized as follows:

(a) From the results of benchmark functions, the PBSPSO algorithm with parallel structure and cooperative mechanism is shown to be effective in search capabilities and results stability.
(b) The employment of a local boundary search process significantly increases the chance of finding global optima near the boundaries of constraints. This local boundary search procedure is suitable for practical engineering problems. The test results of five engineering problems support the above conclusion.
(c) By combining velocity reset operator with the PSO, the proposed method overcomes the limitation of existing modified constrained PSO algorithms in falling into local solutions. As a result, the global search capability of the PBSPSO algorithm is improved effectively.

In future work, the selection strategy of initial points picking will be further studied and the local boundary search process will be further improved.

Acknowledgements The research leading to the above results was supported by National Natural Science Foundation of China (Grant No. 11772191), National Science Foundation for Young Scientists of China (Grant No. 51705312) and National Postdoctoral Foundation of China (Grant No. 17Z102060055). The authors also acknowledge the support from the Adjunct Professor position provided by the Shanghai Jiao Tong University to Prof. Wei Chen.

Appendix 1 Tension/compression spring design problem

$$f(X) = (x_3 + 2)x_2x_1^2 \qquad \text{(Minimize)}$$

subject to:

$$g_1(X) = 1 - \frac{x_2^3 x_3}{7.1785 x_4} \le 0$$
$$g_2(X) = \frac{4x_2^2 - x_1 x_2}{12.566\left(x_2 x_1^3\right) - x_1^4} + \frac{1}{5.108 x_1^2} - 1 \le 0$$
$$g3(X) = \frac{140.45 x_1}{x_2^2 x_3} 1 \le 0$$
$$g_4(X) = \frac{x_2 + x_1}{x_2^2 x_3} - 1 \le 0$$
$$with\, 0.05 \le x_1 \le 2.0,\, 0.25 \le x_2 \le 1.3,$$
$$and\, 2.0 \le x_3 \le 15.0.$$

Appendix 2 Three bar truss design problem

$$f(X) = \left(2\sqrt{2}x_1 + x_2\right) l \qquad \text{(Minimize)}$$

subject to:

$$g_1(X) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1 x_2} P - \sigma \le 0$$
$$g_2(X) = \frac{2}{\sqrt{2}x_1^2 + 2x_1 x_2} P - \sigma \le 0$$
$$g_3(X) = \frac{1}{\sqrt{2}x_2 + x_1} P - \sigma \le 0$$
$$0 \le x_i \le 1, i = 1, 2$$
$$l = 100cm, p = 2kN/cm^2, \sigma = 2kN/cm^2$$

Appendix 3 Welded beam design problem

$$f(X) = 1.10471 x_1 x_2^2 + 0.04811 x_3 x_4 (14 + x_2) \qquad \text{(Minimize)}$$

subject to:

$$g_1(X) = \tau(X) - \tau_{\max} \le 0$$
$$g_2(X) = \sigma(X) - \sigma_{\max} \le 0$$
$$g_3(X) = x_1 - x_4 \le 0$$
$$g_4(X) = 0.10471_1^2 + 0.4811 x_3 x_4 (14 + x_2) - 5 \le 0$$
$$g_5(X) = 0.125 - x_1 \le 0$$
$$g_6(X) = \delta(X) - 0.25 \le 0$$
$$g_7(X) = P - P_c(X) \le 0$$
$$0.1 \le x_1, x_4 \le 2$$
$$0.1 \le x_2, x_3 \le 10$$
$$where\, \tau(X) = \sqrt{\tau'^2 + 2\tau' x_2 + 2R + \tau''^2}$$
$$\tau' = \frac{P}{\sqrt{2}x_1 x_2}, \tau'' = \frac{MR}{J}, M = P\left(14\frac{x_2}{2}\right)$$
$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x1 + x3}{2}\right)^2}, J = 2\left\{\sqrt{2}x_1 x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)\right]\right\},$$
$$\sigma(X) = \frac{504000}{x_4 x_3^2}, \delta(X) = \frac{65856000}{\left(30 \times 10^6 x_4 x_3^3\right)}$$
$$Pc = \frac{4.013 \times \left(30 \times 10^6\right)}{196}\sqrt{\frac{x_3^2 x_4^6}{36} \times \left[1 - \left(x_3\sqrt{\frac{30 \times 10^6}{4 \times \left(12 \times 10^6\right)}}/28\right)\right]},$$
$$P = 6000lb, L = 14in, E = 30 \times 10^6 psi, G = 12 \times 10^6 psi,$$
$$\tau_{\max} = 13600psi, \sigma_{max} = 30000psi, \delta_{max} = 0.25in$$

Appendix 4 Pressure vessel design problem

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

subject to:

$$g_1(X) = -x_1 + 0.0193 x_3 \le 0$$
$$g_2(X) = -x_2 + 0.0954 x_3 \le 0$$
$$g_3(X) = -\pi x_3^2 - \left({}^4/_3\right)\pi x_3^3 + 1296000 \le 0$$
$$g_4(X) = x_4 - 240 \le 0$$
$$0 \le x_i \le 100, i = 1, 2$$
$$10 \le x_i \le 200, i = 3, 4$$

Appendix 5 Speed reducer design problem

$$f(X) = 0.7854 x_1 x_2^2\left(3.3333 x_3^2 + 14.9334 x_3 - 43.0934\right) - 1.508 x_1\left(x_6^2 + x_7^2\right) + 7.4777\left(x_6^3 + x_7^3\right) + 0.7854\left(x_4 x_6^2 + x_5 x_7^2\right) \qquad \text{(Minimize)}$$

subject to:

$$g_1(X) = \frac{27}{x_1 x_1^2 x_3} - 1 \le 0$$
$$g_2(X) = \frac{397.5}{x_1 x_2^2 x_3^2} - 1 \le 0$$
$$g_3(X) = \frac{1.93 x_4^3}{x_2 x_3 x_6^4} - 1 \le 0$$
$$g_4(X) = \frac{1.93 x_5^3}{x_2 x_3 x_7^4} - 1 \le 0$$
$$g_5(X) = \frac{1.0}{110 x_6^4}\sqrt{\left(\frac{745.0 x_4}{x_2 x_3}\right) + 16.9 \times 10^6} - 1 \le 0$$
$$g_6(X) = \frac{1.0}{85 x_7^4}\sqrt{\left(\frac{745.0 x_5}{x_2 x_3}\right) + 157.5 \times 10^6} - 1 \le 0$$
$$g_7(X) = \frac{x_2 x_3}{40} - 1 \le 0$$
$$g_8(X) = \frac{5 x_2}{x_1} - 1 \le 0$$
$$g_9(X) = \frac{x_1}{12 x_2} - 1 \le 0$$
$$g_{10}(X) = \frac{1.5 x_6 + 1.9}{x_4} - 1 \le 0$$
$$g_{11}(X) = \frac{1.1 x_7 + 1.9}{x_5} - 1 \le 0$$
$$with\, 2.6 \le x_1 \le 28, 0.7 \le x_2 \le 0.8,$$
$$17 \le x_3 \le 28, 7.3 \le x_4 \le 8.3, 7.3 \le x_5 \le 8.3,$$
$$2.9 \le x_6 3.9, 5.0 \le x_7 \le 5.5,$$

References

Bonyadi MR, Michalewicz Z (2014) On the edge of feasibility: a case study of the particle swarm optimizer. Ieee Congress on Evolutionary Computation. p. 3059–66

Kennedy J, Eberhart R (1995) Particle swarm optimization, in: 1995 I.E. international conference on neural networks proceedings, pp. 1942–1948(vol 48)

Parsopoulos KE, Vrahatis MN (2002) Particle Swarm Optimization Method for Constrained Optimization Problem: CiteSeer,

Pulido GT, Coello CAC (2004) A constraint-handling mechanism for particle swarm optimization. Cec2004: Proceedings of the 2004 Congress on Evolutionary Computation. p. 1396–403 Vol.2

Hu XH, Eberhart R (2002) Solving constrained nonlinear optimization problems with particle swarm optimization, in: 6th world multi-conference on Systemics, cybernetics and informatics (SCI 2002)/ 8th international conference on information systems analysis and synthesis (ISAS 2002)

Guo CX, Hu JS, Ye B, Cao YJ (2004) Swarm intelligence for mixed-variable design optimization. J Zhejiang Univ (Sci) 5:851–860

He Q, Wang L (2007a) An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Eng Appl Artif Intell 20:89–99

He Q, Wang L (2007b) A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. Appl Math Comput 186:1407–1422

Paquet U, Engelbrecht AP (2007) Particle swarms for linearly constrained optimisation. Fundamenta Informaticae 76:147–170

Liang JJ, Shang Z, Li Z (2010) Coevolutionary comprehensive learning particle swarm optimizer. IEEE Congress on Evolutionary Computation:1–8

C-l S, J-c Z, J-s P (2011) An improved vector particle swarm optimization for constrained optimization problems. Inf Sci 181:1153–1163

Bonyadi M, Li X, Michalewicz Z (2013) A hybrid particle swarm with velocity mutation for constraint optimization problems. Proceeding of the Fifteenth Conference on Genetic and Evolutionary Computation Conference. p. 1–8

Mezura-Montes E, Coello Coello CA (2011) Constraint-handling in nature-inspired numerical optimization: past, present and future. Swarm and Evolutionary Computation 1:173–194

Shi Y, Eberhart R (1999) Modified particle swarm optimizer. IEEE international conference on evolutionary computation proceedings, 1998 I.E. world congress on Computational Intelligence 1999. p. 69–73

Kennedy J (1997) The particle swarm: social adaptation of knowledge. IEEE International Conference on Evolutionary Computation 1997. p. 303–8

Eberhart RC, Shi YH (2001) Particle swarm optimization: Developments, applications and resources,

Yang JM, Chen YP, Horng JT, Kao CY (1997) Applying family competition to evolution strategies for constrained optimization. Evolutionary Programming Vi, International Conference, Ep97, Indianapolis, Indiana, Usa, April 13-16, , Proceedings1997. P. 201–11

Liu Z, Lu J, Zhu P (2016) Lightweight design of automotive composite bumper system using modified particle swarm optimizer. Compos Struct 140:630–643

Clerc M (2006) Confinements and biases in particle swarm optimisation

Liang JJ (2006) PNS. Problem Definitions and Evaluation criteria for the CEC 2006 special session on constrained real-parameter Optimization

Mezura-Montes E, Coello CACA (2005) Simple multimembered evolution strategy to solve constrained optimization problems. IEEE Trans Evol Comput 9:1–17

Wang Y, Cai Z, Guo G, Zhou Y (2007) Multiobjective optimization and hybrid evolutionary algorithm to solve constrained optimization problems. Ieee Transactions on Systems Man and Cybernetics Part B-Cybernetics 37:560–575

Wang Y, Cai Z, Zhou Y, Zeng W (2008) An adaptive tradeoff model for constrained evolutionary optimization. IEEE Trans Evol Comput 12:80–92

Mezura-Montes E, Cetina-Dominguez O (2012) Empirical analysis of a modified artificial bee Colony for constrained numerical optimization. Appl Math Comput 218:10943–10973

Dhadwal MK, Jung SN, Kim CJ (2014) Advanced particle swarm assisted genetic algorithm for constrained optimization problems. Comput Optim Appl 58:781–806

Venter G, Haftka RT (2010) Constrained particle swarm optimization using a bi-objective formulation. Struct Multidiscip Optim 40:65–76

Zhang M, Luo W, Wang X (2008) Differential evolution with dynamic stochastic selection for constrained optimization. Inf Sci 178:3043–3074

Garcia S, Molina D, Lozano M, Herrera F (2009) A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization. J Heuristics 15:617–644

Wang Y, Cai Z, Zhou Y, Fan Z (2009) Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique. Struct Multidiscip Optim 37:395–413

Eskandar H, Sadollah A, Bahreininejad A, Hamdi M (2012a) Water cycle algorithm - a novel metaheuristic optimization method for solving constrained engineering optimization problems. Comput Struct 110: 151–166

Wang L, Li LP (2010) An effective differential evolution with level comparison for constrained engineering design. Struct Multidiscip Optim 41:947–963

Coelho LDS (2010) Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems. Expert Syst Appl 37:1676–1683

Sadollah A, Bahreininejad A, Eskandar H, Hamdi M (2013) Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems. Appl Soft Comput 13:2592–2612

Zahara E, Kao YT (2009) Hybrid Nelder–mead simplex search and particle swarm optimization for constrained engineering design problems. Expert Syst Appl 36:3880–3886

Guedria NB (2016) Improved accelerated PSO algorithm for mechanical engineering optimization problems. Appl Soft Comput 2016:455–467

Coello CAC, Montes EM (2002) Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. Adv Eng Inform 16:193–203

Coello CAC, Becerra RL (2004) Efficient evolutionary optimization through the use of a cultural algorithm. Eng Optim 36:219–236

Lampinen J (2002) A constraint handling approach for the differential evolution algorithm. Cec'02: proceedings of the 2002 congress on evolutionary computation, Vols 1 and 2, . p. 1468–73

Liu H, Cai Z, Wang Y (2010) Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. Appl Soft Comput 10:629–640

Kashan AH (2011) An efficient algorithm for constrained global optimization and application to mechanical engineering design: league championship algorithm (LCA): Butterworth-Heinemann

Yang XS (2010) Engineering optimization—an introduction with metaheuristic applications. New York. John Wiley & Sons, Wiley, pp 291–298

Cuevas E, Cienfuegos M (2014) A new algorithm inspired in the behavior of the social-spider for constrained optimization. Expert Syst Appl 41:412–425

Ray T, Liew KM (2003) Society and civilization: an optimization algorithm based on the simulation of social behavior. IEEE Trans Evol Comput 7:386–396

Eskandar H, Sadollah A, Bahreininejad A, Hamdi M (2012b) Water cycle algorithm &ndash; a novel metaheuristic optimization method for solving constrained engineering optimization problems. Comput Struct:151–166

Gandomi AH, Yang X-S, Alavi AH (2011) Mixed variable structural optimization using firefly algorithm. Comput Struct 89:2325–2336

He Q, Wang L (2007c) A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. Appl Math Comput 186:1407–1422

Fz H, Wang L, He Q (2007) An effective co-evolutionary differential evolution for constrained optimization. Appl Math Comput 186: 340–356

Akay B, Karaboga D (2012) Artificial bee colony algorithm for large-scale problems and engineering design optimization. J Intell Manuf 23:1001–1014