Advances in Nonlinear Dynamics and Vibrations on Mechanical Systems - Research Article

Advances in Mechanical Engineering

Advances in Mechanical Engineering
2019, Vol. 11(3) 1–27

DOI: 10.1177/1687814018824930
journals.sagepub.com/home/ade
SAGE

A dynamic adaptive particle swarm optimization and genetic algorithm for different constrained engineering design optimization problems

Hao Zhu[1,2], Yumei Hu[3] and Weidong Zhu[4]

Abstract

A dynamic adaptive particle swarm optimization and genetic algorithm is presented to solve constrained engineering optimization problems. A dynamic adaptive inertia factor is introduced in the basic particle swarm optimization algorithm to balance the convergence rate and global optima search ability by adaptively adjusting searching velocity during search process. Genetic algorithm–related operators including a selection operator with time-varying selection probability, crossover operator, and *n*-point random mutation operator are incorporated in the particle swarm optimization algorithm to further exploit optimal solutions generated by the particle swarm optimization algorithm. These operators are used to diversify the swarm and prevent premature convergence. Tests on nine constrained mechanical engineering design optimization problems with different kinds of objective functions, constraints, and design variables in nature demonstrate the superiority of the dynamic adaptive particle swarm optimization and genetic algorithm against several other meta-heuristic algorithms in terms of solution quality, robustness, and convergence rate in most cases.

Keywords

Constrained engineering design optimization problems, continuous and discrete design variables, meta-heuristic, dynamic adaptive, particle swarm optimization, genetic algorithm

Date received: 18 August 2018; accepted: 12 December 2018

Handling Editor: Yunn-Lin Hwang

Introduction

A great number of optimization algorithms have been proposed to solve different engineering design optimization problems which are usually nonlinearly constrained ones. The optimization algorithms can be roughly divided into two categories: a stochastic algorithm and deterministic one. The traditional deterministic optimization methods, such as the steepest descend method, quasi-Newton method, and interior-reflective Newton method, are usually gradient-based algorithms and differentiable conditions of objective functions are required to meet. These methods are inefficient and inaccurate for complex optimization problems with strong nonlinearity and high dimensions especially when the objective functions and constraints are discontinuous and not smooth.[1] Numerous

[1]Department of Mechanics and Engineering Science, Sichuan University, Chengdu, China
[2]Key Laboratory of Deep Underground Science and Engineering (Ministry of Education), School of Architecture and Environment, Sichuan University, Chengdu, China
[3]State Key Laboratory of Mechanical Transmissions, Chongqing University, Chongqing, China
[4]Department of Mechanical Engineering, University of Maryland, Baltimore County, Baltimore, MD, USA

Corresponding author:
Hao Zhu, Department of Mechanics and Engineering Science, Sichuan University, Chengdu 610065, China.
Email: haozhu@scu.edu.cn

stochastic optimization algorithms, such as the particle swarm optimization (PSO) algorithm,[2] genetic algorithm (GA),[3–5] firefly algorithm,[6] ant colony optimization,[7] artificial bee colony (ABC),[8] mine blast algorithm (MBA),[9] simulated annealing (SA) algorithm,[10] biogeography-based optimization (BBO) algorithm[11], have been proposed to overcome these drawbacks. These stochastic optimization algorithms are usually meta-heuristic and inspired by physical and natural phenomena.

Among all these stochastic optimization algorithms, the PSO algorithm is widely applied to solve different engineering optimization problems as it is efficient in computation, easy for implementation, and reliable in searching for global optima.[12–16] The PSO algorithm first proposed by Kennedy and Eberhart[2] is based on social sharing of information between individuals in a group and is originated from mimicking the flocking behavior of a swarm of fish and imitating the schooling behavior of birds. The PSO algorithm is made up of a population of particles which are randomly moving within the parameter space. The position of each individual particle in the parameter space denotes a candidate solution of the design optimization problem. By changing searching velocities and positions of particles, the optimal solution is found. The ability of searching optima of the PSO algorithm mainly relies on mutual interaction (social learning) and influence of individual particles (cognitive learning). Particles move toward the currently global best position of the swarm in each iteration. A particle can escape from a local optimum with the help of neighboring particles. But if most of its neighboring particles are limited to a local extreme point, it is attracted to the trap of the local optimum, and as a result, premature convergence of the algorithm and the stagnation phenomenon[17] occur. To overcome these drawbacks of the basic PSO algorithms, different improvements have been proposed. A descending dynamic inertia factor or accelerating factor is widely adopted to balance the convergence rate and space searching ability of the PSO algorithm during search process.[16,18,19] Eberhart and Shi[20] applied a random inertia weight factor to deal with dynamic systems. Clerc[21] presented a constriction factor K to control the convergence velocity. Apart from using time-varying inertia weights (TVIW), time-varying accelerating coefficients (TVAC) were also proposed and used to control the convergence rate and solution quality.[22,23] A co-evolutionary particle swarm optimization (CPSO) was presented by He and Wang[24] to solve constrained engineering optimization problems. They used a multiple-swarm technique to evolve decision solutions and adapt penalty factors. Later, Krohling and Coelho[25] improved the CPSO by dynamically adjusting the accelerated coefficients which satisfy Gaussian probability distribution. Worasucheep[26] presented a constrained PSO algorithm with the stagnation detection and dispersion mechanism to tackle real word nonlinear and constrained engineering optimization problems. Yang and colleagues[27,28] proposed an accelerated particle swarm optimization (APSO) algorithm based on the basic PSO algorithm, in which the velocity vector is removed and particle best positions are replaced by randomness. This algorithm greatly improves calculation efficiency and implementation convenience. However, this algorithm is easily trapped in premature convergence particularly for the problems with high nonlinearity due to the deficiency of diversity.[1] This disadvantage was improved by Guedria[1] by incorporating memories of individual particles into APSO forming a new algorithm called improved adaptive particle swarm optimization (IAPSO).

To improve the swarm diversity and increase convergence rate, many hybrid optimization algorithms with some operators or other algorithms incorporated into PSO have been proposed.[29–34] Novitasari et al.[29] proposed a hybrid algorithm that combines the SA with PSO algorithm to deal with constrained optimization problems. He and Wang[30] proposed a similar hybrid algorithm to optimize a support vector machine. Wang and Yin[31] introduced a ranking selection scheme into the basic PSO to automatically control search performance of the swarm, which results in a new algorithm called ranking selection–based particle swarm optimization (RSPSO). The crossover operators or mutation operators used in GAs were largely adopted by researchers and combined with PSO to generate new algorithms, such as the modified particle swarm optimization (MPSO),[32] quantum-behaved PSO using mutation operator with Gaussian distribution (G-QPSO),[33] straightforward particle swarm optimization (SPSO) with a logistic chaotic mutation operator,[34] self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients (HPSO-TVAC),[22] and so on. These operators increase swarm diversity and prevent premature convergence and stagnation of the PSO algorithms. The hybrid optimization algorithms talked above have been used to solve different specific engineering optimization problems.

In this work, a dynamic adaptive particle swarm optimization and genetic algorithm (DAPSO-GA) previously proposed by us in Zhu et al.[35] is used to solve constrained engineering design optimization problems with different kinds of design variables. A dynamic adaptive inertia factor is used in the PSO algorithm to adjust its convergence rate and control the balance of global and local optima exploration. GA-related operators including a selection operator with time-varying selection probability, crossover operator, and n-point random mutation operator are incorporated into the PSO to further exploit the optimal solutions generated by the PSO-related algorithm. These operators are used to diversify the swarm and prevent premature convergence. The remainder of this work is organized as bellows. The DAPSO-GA for both continuous and discrete optimization problems with constraints is specifically introduced in section "Introduction of the DAPSO-GA." In section "Constrained engineering optimization problems," four benchmark constrained engineering optimization problems with continuous design variables and five ones with discrete or mixed design variables are used to evaluate performance of the DAPSO-GA on real word engineering optimization problems. Conclusions are drawn in section "Conclusion."

Introduction of the DAPSO-GA

The DAPSO-GA is a hybrid algorithm that combines the GA and PSO algorithm. Specifically, the GA-related operators including selection, crossover, and n-point random mutation operators are incorporated into the PSO algorithm with craft. These GA-related operators are used to diversify the swarm and further explore the possible optima based on the feasible solution provided by the PSO algorithm.

PSO-related algorithm

Basic PSO algorithm. The basic PSO algorithm is made up of a population of particles that are randomly spread within the parameter space. The position of each individual particle in the parameter space denotes a candidate solution of the design optimization problem. Each particle has a velocity and moves in the parameter space. The position and velocity of the particle i are adjusted in each iteration

$$\mathbf{v}_i(\lambda+1) = \omega\,\mathbf{v}_i(\lambda) + c_1 r_1(\mathbf{P}_i(\lambda) - \mathbf{x}_i(\lambda)) + c_2 r_2\big(\mathbf{P}_g(\lambda) - \mathbf{x}_i(\lambda)\big) \tag{1}$$

$$\mathbf{x}_i(\lambda+1) = \mathbf{x}_i(\lambda) + \mathbf{v}_i(\lambda) \tag{2}$$

where $\mathbf{x}_i(\lambda)$ and $\mathbf{v}_i(\lambda)$ are the position and velocity of the particle at time step λ, respectively; $\mathbf{P}_i(\lambda)$ is the historical best position of the particle i so far and $\mathbf{P}_g(\lambda)$ is the global best position of the whole swarm up to a time step λ; r_1 and r_2 are random numbers within a range from 0 to 1; ω is an inertia factor; and c_1 and c_2 are two accelerating factors used to scale influence of the best positions of the particle i and global best position of the swarm, respectively. To ensure convergence of the PSO algorithm, the two accelerating factors are constrained by[13,16]

$$\begin{cases} 0 < (c_1 + c_2) < 4 \\ (c_1 + c_2)/2 - 1 < \omega < 1 \end{cases} \tag{3}$$

The procedure of the basic PSO algorithm begins with population initialization of particles with random positions and velocities. The positions and velocities of each particle are then updated by equations (1) and (2). After that, the corresponding fitness of each particle is evaluated and ranked, and $\mathbf{P}_i(\lambda)$ and $\mathbf{P}_g(\lambda)$ are updated. The above procedure is repeated until an ending criterion is met. The ending criterion is usually the maximum number of iterations or a sufficiently low error bound.

PSO-related algorithm in the DAPSO-GA. A dynamic adaptive inertia factor $\omega_i(\lambda)$ is introduced into the basic PSO to adaptively adjust its searching velocity during iterations

$$\mathbf{v}_i(\lambda+1) = \omega_i(\lambda)\cdot\mathbf{v}_i(\lambda) + c_1 r_1(\mathbf{P}_i(\lambda) - \mathbf{x}_i(\lambda)) + c_2 r_2\big(\mathbf{P}_g(\lambda) - \mathbf{x}_i(\lambda)\big) \tag{4}$$

where

$$\omega_i(\lambda) = \omega_{\min} + (\omega_{\max} - \omega_{\min})\sin\left(\frac{\beta_i(\lambda)\pi}{2}\right) \in [\omega_{\min},\ \omega_{\max}] \tag{5}$$

in which

$$\beta_i(\lambda) = \frac{f_i(\lambda) - f_g(\lambda)}{f_w(\lambda) - f_g(\lambda)} \in [0,\ 1], \quad i = 1,\ 2,\ \ldots,\ N \tag{6}$$

with $f_i(\lambda)$ being the fitness value of the ith particle in the λth iteration, and $f_g(\lambda)$ and $f_w(\lambda)$ being the best and worst fitness values of the swarm in the λth iteration, respectively; and they satisfy $f_g(\lambda) \leqslant f_i(\lambda) \leqslant f_w(\lambda)$ and thus $\beta_i(\lambda) \in [0,\ 1]$. Particles with the best fitness value and worst fitness value are called the best particle and worst particle in the swarm, respectively. From equations (5) and (6), the inertia factor is adaptively adjusted in the range $[\omega_{\min},\ \omega_{\max}]$ during iteration. The better fitness value a particle has, the smaller the inertia factor is. Large inertia factor represents a large searching velocity and thus, more solution spaces will be explored. In contrast, small inertia factor can help the PSO algorithm further exploit the solution space around the best particle. Hence, this dynamic adjustment of the inertia factor can adaptively balance the convergence rate and global optima search ability of the PSO algorithm.

Each particle position $\mathbf{x}_i$ is limited in the range $[\mathbf{x}_{\min},\ \mathbf{x}_{\max}]$. If $\mathbf{x}_i$ locates outside this range, it will be replaced by

$$x_i^k = \begin{cases} x_{\max}^k, & if\ x_i^k > x_{\max}^k \\ x_{\min}^k, & if\ x_i^k < x_{\min}^k \end{cases},\ k = 1,\ 2,\ \ldots,\ D \tag{7}$$

in which D is the particle dimension and x_i^k is the position of the ith particle in the kth dimension. Each particle velocity $\mathbf{v}_i(\lambda)$ is limited in $[\mathbf{v}_{\min},\ \mathbf{v}_{\max}]$, in which

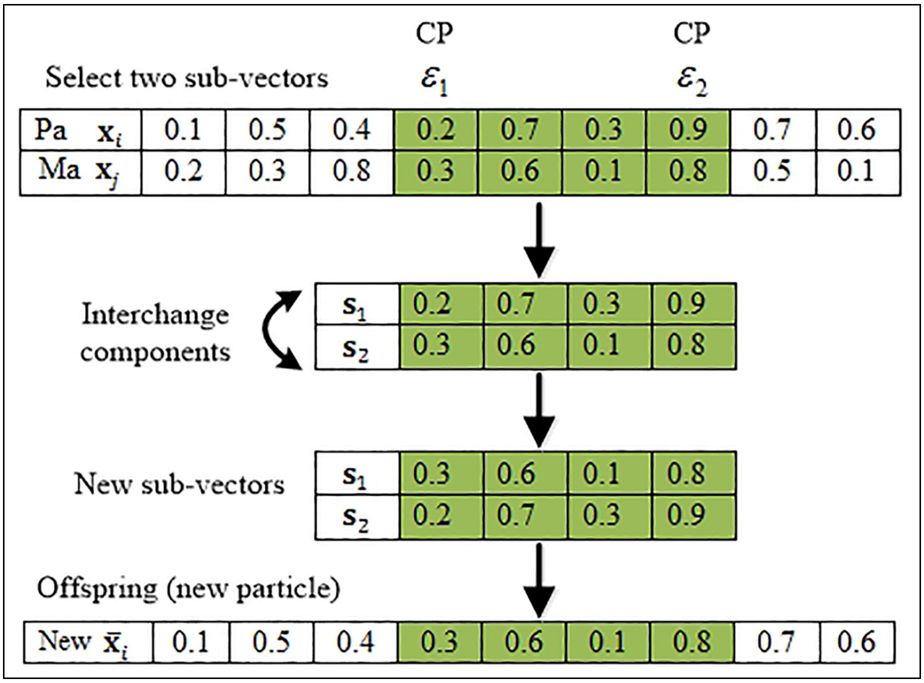

Figure 1. Flowchart of the crossover operator of the GA-related algorithm in the DAPSO-GA.

$\mathbf{v}_{\min} = -(\mathbf{x}_{\max} - \mathbf{x}_{\min})/2$ and $\mathbf{v}_{\max} = (\mathbf{x}_{\max} - \mathbf{x}_{\min})/2$. If the particle velocity violates this limit, it will be replaced by

$$v_i^k = \begin{cases} v_{\max}^k, & if\ v_i^k > v_{\max}^k \\ v_{\min}^k, & if\ v_i^k < v_{\min}^k \end{cases}, \quad k = 1, 2, \ldots, D \tag{8}$$

in which v_i^k is the velocity of the ith particle in the kth dimension.

GA-related algorithm

In the DAPSO-GA, GA-related operators, that is, the selection operator with time-varying selection probability, crossover operator, and n-point random mutation operator are introduced to further exploit the optimal solutions generated by the adaptive PSO algorithm. GA uses a population which consists of individuals or chromosomes and each individual stands for a potential solution. In the GA-related algorithm, each particle in the swarm is regarded as an individual or chromosome and the swarm constitutes a population. Each individual is represented by applying decimal coding (the real value).

Adaptive dynamic selection operator. A particle that meets the GA-selection criterion below is selected to update its position via the following crossover and mutation operators in iteration

$$0 < \left| \frac{f_i(\lambda) - f_g(\lambda)}{f_g(\lambda)} \right| < \eta \tag{9}$$

where $f_i(\lambda)$ is the current fitness value of the ith particle at the λth iteration, $f_g(\lambda)$ is the best fitness value of the swarm that corresponds to its global best position, and $\eta = \eta_{\max} - \lambda(\eta_{\max} - \eta_{\min})/\lambda_{tot}$ is the time-varying selection probability which descends from $\eta_{\max}$ to $\eta_{\min}$ during iteration process.

Crossover and mutation operator. When the GA-selection criterion is met, the following two GA-related operators are used to update the particle position: randomly generate a number $\alpha \in [0, 1]$, and then a crossover operator is applied if $\alpha \leqslant P_c$, where P_c is predefined crossover probability; otherwise, an n-point random mutation operator is applied and $1 - P_c$ is corresponding mutation probability.

Crossover operator. A random crossover operator is adopted here to generate a new individual (particle). The flowchart of the crossover operator is illustrated in Figure 1. First, two particles should be selected as parents (pa and ma) for breeding. Suppose the ith particle is already selected as pa according to the GA-selection criterion, and then another jth particle is randomly selected as ma from the swarm, where $j \in [1, M]$ and $j \neq i$. Second, two cutting points (CPs), that is, ε_1 and ε_2, are randomly generated and then two sub-vectors

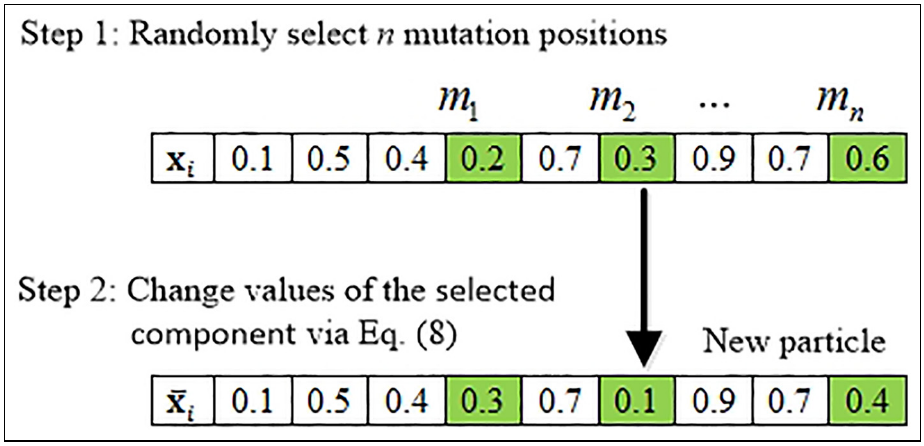

Figure 2. Procedures of the *n*-point mutation operator.

($\mathbf{s}_1$ and $\mathbf{s}_2$) are picked out from the position vectors of the parents, where $0 < \varepsilon_1 \leqslant \varepsilon_2 \leqslant D$. Components of the selected two sub-vectors are inter-changed and then a new particle (offspring) is generated.

Mutation operator. An n-point random mutation operator is used, where n is the mutation dimension (i.e. the number of components or genes of the selected particle or chromosome for mutation) which is a random integer in $[1, D]$. It means that there are in total n points (genes) in the selected particle (chromosome) to be changed via mutation. Procedure of the n-point random mutation operator is shown in Figure 2. First, the mutation dimension n of the selected particle is identified by $n = round(rand \times D)$, in which $rand$ is a random number in [0, 1] and $round$ is an operator to round off the product of $rand$ and D. Second, n different integers (i.e. $m_1, m_2, \ldots, m_n$), which are limited in the range $[1, D]$, are randomly generated. These integers represent the mutation positions in the position vector of the selected particle. Next, values of the components of the selected particle position vector are randomly changed via the following equation

$$\bar{x}_i^k = x_{\min}^k + \sin\left(rand \times \frac{\pi}{2}\right) \times \left(x_{\max}^k - x_{\min}^k\right),\ k = m_1, m_2, \ldots, m_n \tag{10}$$

Implementation procedure of the DAPSO-GA algorithm

Flowchart of the DAPSO-GA is shown in Figure 3 and it is briefly described as follows:

Step 1: Set initial values of the optimization parameters including the population size M, maximum number of generations (iterations) S, maximum and minimum inertia factors $\omega_{\max}$ and $\omega_{\min}$, respectively, accelerating factors c_1 and c_2, maximum and minimum selection probability $\eta_{\max}$ and $\eta_{\min}$, respectively, crossover probability p_c, upper and lower limits of the position of each particle $\mathbf{x}_i^u$ and $\mathbf{x}_i^l$, respectively. In this work, $\omega_{\max} = 0.7$, $\omega_{\min} = 0.4$, $\eta_{\max} = 0.7$, $\eta_{\min} = 0.15$, $c_1 = c_2 = 2$, and $p_c = 0.5$ are used.
Step 2: Initialize the swarm: randomly generate a swarm with a size of M and the initial position of each particle is given by

$$\mathbf{x}_i(0) = \mathbf{x}_i^l + rand\left(\mathbf{x}_i^u - \mathbf{x}_i^l\right),\ i = 1, 2, \ldots, M \tag{11}$$

Step 3: Evaluate the fitness value of each initially generated particle and rank their positions. The initial best particle position $\mathbf{P}_i(0)$ and initial global best and worst positions $\mathbf{P}_g(0)$ and $\mathbf{P}_w(0)$ of the swarm, respectively, are then identified.
Step 4: Update the current position $\mathbf{x}_i(\lambda)$ and velocity $\mathbf{v}_i(\lambda)$ of the ith particle according to equations (2) and (4)–(6).
Step 5: Evaluate the current fitness value of each particle, and update the best particle position $\mathbf{P}_i(\lambda)$ and global best and worst positions of the swarm $\mathbf{P}_g(\lambda)$ and $\mathbf{P}_w(\lambda)$, respectively.
Step 6: Generate new particles (offspring) according to the GA-related algorithm to diversify the swarm. If the GA-selection criterion in equation (9) is met, the crossover operator and n-point random mutation operator are applied to update the position of a selected particle to generate a new particle $\mathbf{x}_i(\lambda)$ as presented in section "Crossover and mutation operator."
Step 7: Evaluate the fitness value of the new particle $\bar{f}_i(\lambda)$ and compare it with the best and worst fitness values of the swarm $f_g(\lambda)$ and $f_w(\lambda)$, respectively. If

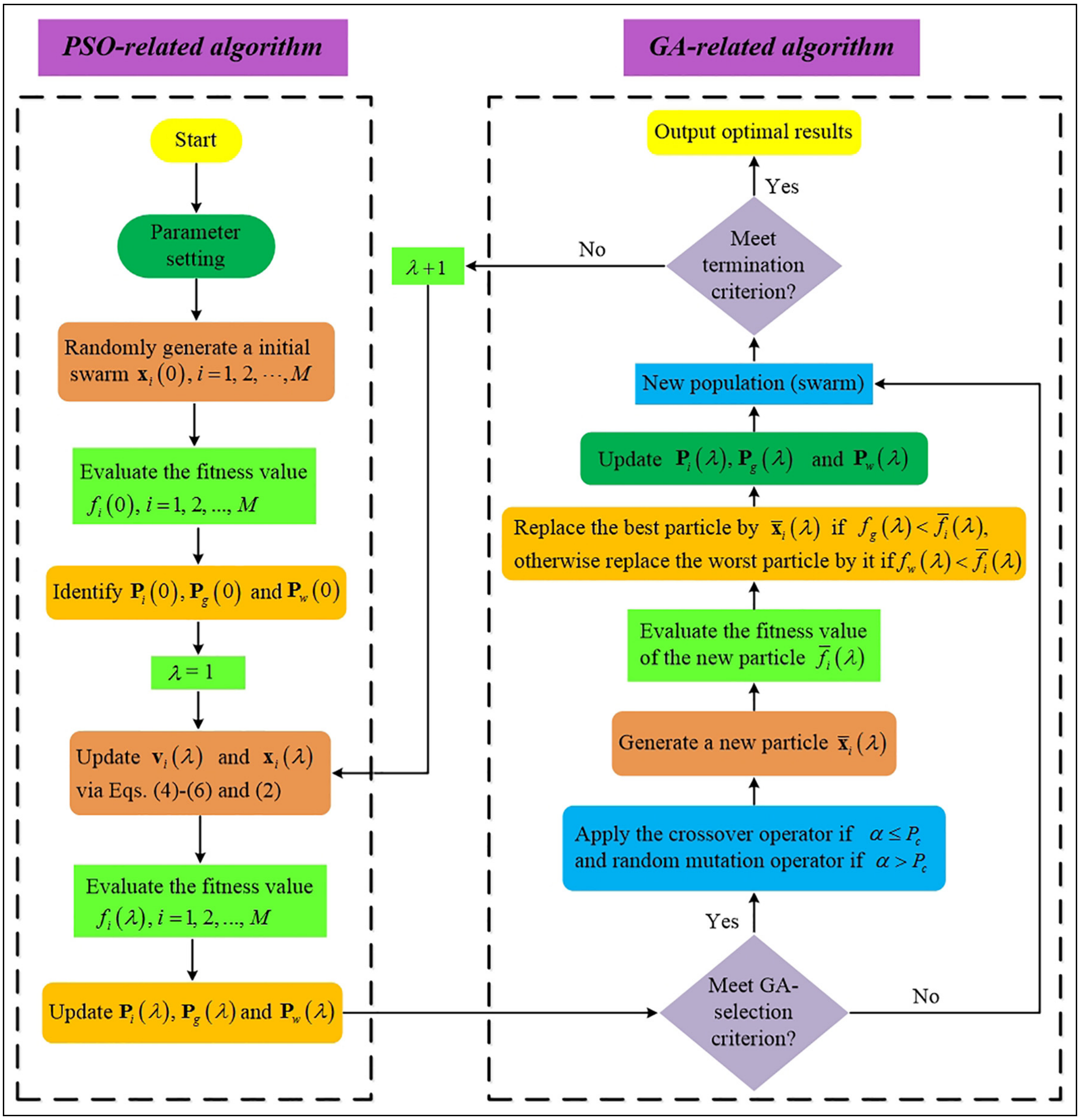

Figure 3. Flowchart of the DAPSO-GA.

$\bar{f}_i(\lambda) < f_g(\lambda)$, replace the best particle by it; otherwise, replace the worst particle by it if $\bar{f}_i(\lambda) < f_w(\lambda)$. Update the best particle position $\mathbf{P}_i(\lambda)$ and global best and worst positions of the swarm $\mathbf{P}_g(\lambda)$ and $\mathbf{P}_w(\lambda)$ if necessary.

Step 8: Repeat the above steps 4–7 until the termination criterion, which is a predefined number of iteration, is met and then output the optimal results.

Strategies of the DAPSO-GA for discrete optimization problems

The DAPSO-GA talked above is suitable for a continuous optimization problem, but cannot handle the optimization problems with discrete variables. For the discrete optimization problems, the DAPSO-GA can be modified using the rounding off approach. In this approach, either the continuous or discrete variables are treated as continuous variables during optimization processes. Only at the end of the optimization procedure, the discrete variables will be rounded off to evaluate the fitness value of each particle as shown below

$$f_i(\mathbf{x}_i, \lambda) = f_i(round(\mathbf{x}_i), \lambda) \tag{12}$$

Values of the discrete variables are in fact not changed as seen in equation (12) and keep unchanged until at the end of each generation of iteration. For convenient description, the DAPSO-GA using the rounding

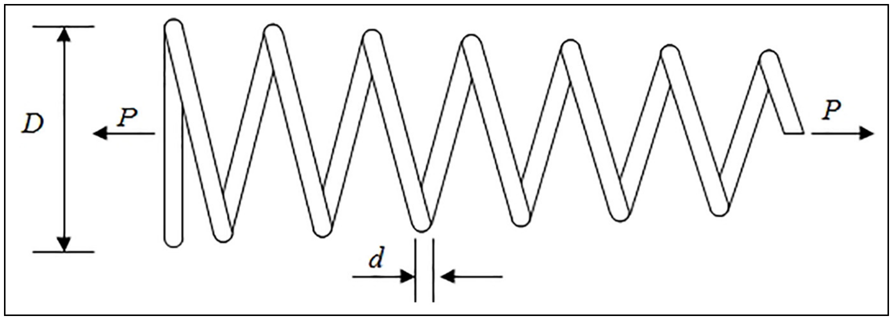

Figure 4. Schematic of the tension/compression spring.

off approach is called a discrete DAPSO-GA and is used to solve the discrete optimization problems later.

Constraints handling

For constrained optimization problems, a feasible solution should satisfy all boundary constraints in the form of the equalities and/or inequalities. Two strategies are used in this work to handle the constraints on design variables and problem-specific constraints. In the DAPSO-GA, each particle position will be reset to the maximum or minimum boundary value once the limits on design variables are violated. Global optima usually occur on or near the boundary of the solution (design) space for the majority of design optimization problems.[9] Hence, this strategy can increase the probability for finding global optimal solutions. Penalty function strategies such as the penalty factor method[1,35–38] and the concept of parameter free penalty function[39,40] are widely used to solve different constrained optimization problems. The penalty factor method is adopted in this work to handle the problem-specific constraints. The constrained optimization problem using the penalty function strategy can be described as bellow

$$\text{Minimize} \quad f_p(\mathbf{x}) = f(\mathbf{x}) + \beta\left(\sum_{j=1}^{N} \mu_j |g_j(\mathbf{x})| + \sum_{m=1}^{L} |h_m(\mathbf{x})|\right) \tag{13}$$

where $f(\mathbf{x})$ and $f_p(\mathbf{x})$ are original and penalized objective functions, respectively; N and L are the total number of inequality constraints and equality constraints, respectively; $g_j(\mathbf{x})$ and $h_m(\mathbf{x})$ are the jth inequality constraint and mth equality constraint, respectively, and

$$\begin{cases} \mu_j = 1, & \textit{if the constraint } g_j(\mathbf{x}) \textit{ is violated} \\ \mu_j = 0, & \textit{else} \end{cases} \tag{14}$$

β is the penalty factor which is a large positive constant that satisfies $\beta \gg f(\mathbf{x})$ and $\beta = 10^{20}$ is adopted in later application in this work. By introducing the penalty term, the constrained optimization problem becomes an unconstrained one as seen in equation (13).

Constrained engineering optimization problems

In this section, nine famous constrained benchmark mechanical engineering optimization problems which have different objective functions, design variables and constraints in nature are adopted to test the performance of the proposed DAPSO-GA in terms of solution quality and stability as well as convergence rate. These 10 constrained engineering optimization problems are divided into continuous and discrete optimization problems according to the categories of their variables, and the rounding off strategy talked in section "Strategies of the DAPSO-GA for discrete optimization problems" is used in the DAPSO-GA to deal with the discrete optimization problems. Statistical results and best solutions of all algorithms for these engineering optimization problems are obtained over 30 independent runs.

Constrained engineering optimization problems with continuous variables

Tension/compression spring design problem. Figure 4 shows a schematic of a tension/compression spring.[41] The design aim of the tension/compression problem (i.e. the objective function $f(\mathbf{x})$) is to minimize its weight with constraints on minimum deformation, shear stress, surge frequency, and maximum outside diameter. These constraints constitute four nonlinear inequality equations as detailed in Appendix 1 (section "Tension/compression spring design problem"). The design problem has three design variables including the wire diameter d, mean coil diameter D, and number of active coils P, which are denoted by x_1, x_2, and x_3 in the objective function and constraint functions, respectively. The DAPSO-GA proposed is used to solve this optimization problem. The swarm size and maximum iteration number are 10 and 200, respectively. Figure 5 presents the convergence history of GA, standard PSO, and the proposed DAPSO-GA for the tension/compression spring problem. It is seen that the standard PSO and DAPSO-GA convergence faster than GA, while the

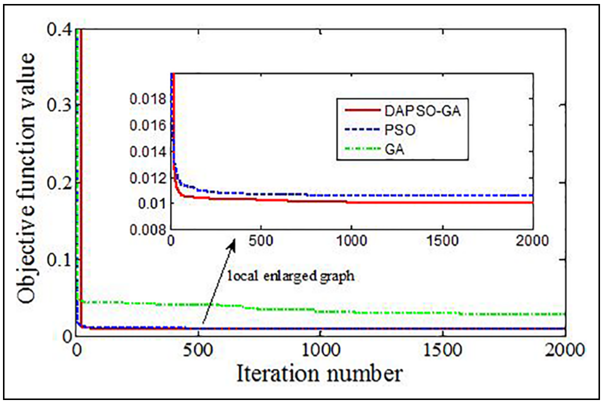

Figure 5. Convergence history of GA, standard PSO, and the proposed DAPSO-GA for the tension/compression spring design problem.

DAPSO-GA has better global optimum searching ability. The DAPSO-GA is also compared with several other meta-heuristic optimization algorithms including the APSO,[27] IAPSO,[1] MBA,[9] ABC2,[40] GA1,[41] GA2,[42] water cycle algorithm (WCA),[43] differential evolution (DE),[44] differential evolution with level comparison (DELC),[45] Nelder-Mead and Particle Swarm Optimization (NM-PSO),[46] hybrid evolutionary algorithm and adaptive constraint handling technique (HEAA),[47] differential evolution with dynamic stochastic selection (DEDS),[36] quantum-behaved particle swarm optimization (QPSO),[48] G-QPSO,[48] society and civilization (SC),[49] league championship algorithm (LCA),[50] cultural algorithms with evolutionary programming (CAEP),[9] unified particle swarm optimization (UPSO),[51] $(\mu + \lambda) - \mathrm{ES}$,[52] and PSO-DE.[53] The optimal solutions obtained by the proposed DAPSO-GA and above optimization algorithms are listed in Table 1. It is apparently seen that the proposed DAPSO-GA finds the best solution with the objective function value 0.009872 that was not found by previously proposed algorithms. Statistical optimization results of all algorithms are listed in Table 2. As seen from Table 2, the DAPSO-GA provides the best solution with least NFEs 2000. HEAA has the best robustness in terms of providing optimal solutions with standard deviation (SD) value of only 1.4E–9 for the tension/compression spring design problem. The PSO algorithm provides the worst solution (0.012857) with the largest SD value and GA1 requires the highest NFEs (900,000).

Figure 6 shows the inertia weight versus number of iterations of the DAPSO-GA on the tension/compression spring design problem. From Figure 6, the inertia weighting factor varies between 0.7 and 0.4. A large inertia weighting factor is used when the fitness value of a particle is far away from the global best fitness value; otherwise, a small one is used. The dynamic inertia weighting factor adaptively adjusts the search velocity so that the exploitation and exploration are well balanced.

Symmetric three-bar truss design problem. Figure 7 presents the schematic diagram of a symmetric three-bar truss structure. The symmetric three-bar truss structure is made up of steel and is subjected to two constant loadings $P_1 = P_2 = P$. The optimization design problem of the three-bar truss structure, which was described by Ray and Liew,[49] is to minimize the volume subject to stress constraints as detailed in Appendix 1 (section "Symmetric three-bar truss design problem"). The design variables are cross-sectional areas of the three bars: x_1, x_2, and x_3. The DAPSO-GA with a swarm size of 20 and maximum number of iterations of 5000 is used to solve this optimization problem. The optimal solution obtained by this algorithm is compared with those obtained by other optimization algorithms such as Hernendez,[54] dynamic stochastic selection for multi-member differential evolution (DSS-MDE),[36] SC,[49]

Table 1. Comparison of optimal solutions obtained from different optimization algorithms for tension/compression spring design problem.

DV	x_1	x_2	x_3	$g_1(\mathbf{x})$	$g_2(\mathbf{x})$	$g_3(\mathbf{x})$	$g_4(\mathbf{x})$	$f(\mathbf{x})$
IAPSO	0.051685	0.356629	11.294175	−1.97E–10	−4.64E–10	−4.05361	−1.091686	0.01266523
APSO	0.052588	0.378343	10.138862	−1.549E–4	−8.328E–4	−4.089171	−1.069069	0.0127
MBA	0.051656	0.35594	11.344665	−0.0009	−0.1344	−4.052248	−0.728268	0.012665
GA1	0.051480	0.351661	11.632201	−2.08E–3	−1.10E–4	−4.026318	−4.026318	0.0127047834
WCA	0.051680	0.356522	11.30041	−1.65E–13	−7.9E–14	−4.053399	−0.727864	0.012665
DELC	0.051689	0.356717	11.288965	−3.4E–9	2.44E–9	−4.053785	−0.727728	0.012665
NM-PSO	0.051620	0.355498	11.333272	1.01E–3	9.94E–4	−4.061859	−0.728588	0.012630
HEAA	0.051689	0.356729	11.288293	3.96E–10	−3.59E–10	−4.053808	−0.72772	0.012665
DEDS	0.051689	0.356717	11.288965	1.45E–9	−1.19E–9	−4.053785	−0.727728	0.012665
G-QPSO	0.051515	0.352529	11.538862	−4.83E–5	−3.57E–5	−4.0455	−0.73064	0.012665
ABC2	0.051689	0.356720	11.288832	−2.53E–13	−5.76E–13	−4.05378	−0.7277	0.012665
DAPSO-GA	0.050	0.3744328	8.54657332	−1.2581E–8	−1.4491E–7	−4.860733	−0.717045	**0.0098724562**

IAPSO: improved adaptive particle swarm optimization; APSO: accelerated particle swarm optimization; DV: design variable; G-QPSO: quantum-behaved PSO using mutation operator with Gaussian distribution; MBA: mine blast algorithm; DAPSO-GA: dynamic adaptive particle swarm optimization and genetic algorithm.

Note: The boldfaced data in each table mean the best one among all the results provided by different algorithms.

Table 2. Comparison of statistical results obtained from different optimization algorithms for tension/compression spring design problem.

Algorithms	Worst	Mean	Best	SD	NFEs
IAPSO	0.01782864	0.013676527	0.01266523	1.57E–3	2000
APSO	0.014937	0.013297	0.012700	6.85E–4	120,000
MBA	0.012900	0.012713	0.012665	6.3E–5	7650
LCA	0.01266667	0.01266541	0.01266523	3.88E–7	15,000
WCA	0.012952	0.012746	0.012665	8.06E–5	11,750
SC	0.016717	0.012922	0.012669	5.9E–4	25,167
PSO-DE	0.012665	0.012665	0.012665	1.2E–8	24,950
HEAA	0.012665	0.012665	0.012665	1.4E–9	24,000
DEDS	0.012738	0.012669	0.012665	1.3E–5	24,000
DELC	0.012665	0.012665	0.012665	1.3E–7	20,000
DE	0.012790	0.012703	0.012670	2.7E–5	204,800
PSO	0.071802	0.019555	0.012857	1.1662E–2	2000
QPSO	0.018127	0.013854	0.012669	1.341E–3	2000
G-QPSO	0.017759	0.013524	0.012669	1.268E–3	2000
NM-PSO	0.012633	0.012631	0.012630	8.47E–7	80,000
HPSO	0.012719	0.012707	0.012665	1.58E–5	81,000
CPSO	0.012924	0.012730	0.012674	5.2E–4	240,000
CAEP	0.015116	0.013568	0.012721	8.42E–4	50,020
GA1	0.012822	0.012769	0.012704	3.94E–5	900,000
GA2	0.012973	0.012742	0.012681	5.9E–5	80,000
UPSO	–	0.02294	0.01312	7.20E–03	100,000
$(\mu+\lambda)-$ES	–	0.013165	0.012689	3.9E–04	30,000
ABC2	0.012710407	0.01266897	0.01266523	9.43E–06	–
DAPSO-GA	**0.015354687**	**0.0107**	**0.0098724562**	**1.591E–3**	**2000**

SD: standard deviation; IAPSO: improved adaptive particle swarm optimization; APSO: accelerated particle swarm optimization; MBA: mine blast algorithm; DAPSO-GA: dynamic adaptive particle swarm optimization and genetic algorithm; PSO: particle swarm optimization; CPSO: co-evolutionary particle swarm optimization; NFE: number of function evaluation; G-QPSO: quantum-behaved PSO using mutation operator with Gaussian distribution; HPSO: hybrid particle optimization algorithm; PSO-DE: Particle swarm optimization with differential optimization.
Note: The boldfaced data mean optimal results provided by the DAPSO-GA algorithm.

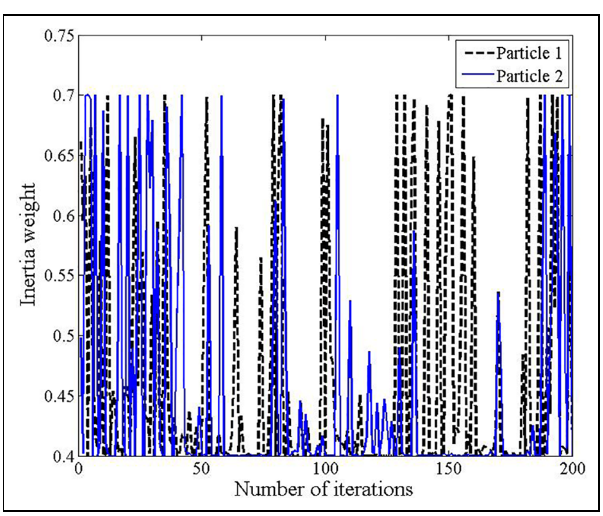

Figure 6. Inertia weight versus number of iterations.

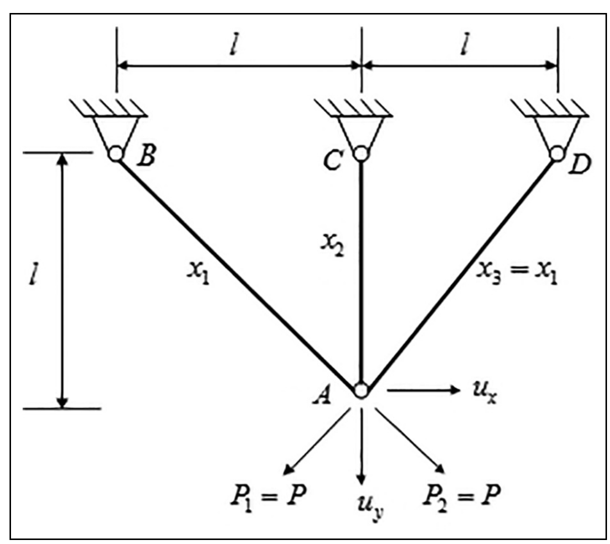

Figure 7. Schematic diagram of the three-bar truss.

swarm with an intelligent information sharing (SIIS),[55] and PSO-TVAC[22] as seen in Table 3.Table 4 presents the comparison of statistical results obtained from these optimization algorithms for the three-bar truss design problem in terms of the worst, mean, and best solutions as well as the SD values and NFEs. As seen from Tables 3 and 4, almost all optimization algorithms provide similar optimal solutions. The proposed DAPSO-GA provides the best solution with the minimum SD value. DAPSO-GA and PSO-TVAC convergence to the best solution with similar NFEs which is less than those

Table 3. Comparison of optimal solutions obtained from different optimization algorithms for the three-bar truss design problem.

DV	Hernendez	DSS-MDE	SC	SIIS	PSO-TVAC	DAPSO-GA
x_1	0.788	0.7886751359	0.788621037	0.795	0.7887058767	0.7886769887
x_2	0.408	0.4082482868	0.408401334	0.395	0.4081613457	0.4082430493
$g_1(\mathbf{x})$	1.637E–3	−2.104E–11	−8.275E–9	−3.376E–3	−4.448E–13	−2.3114E–09
$g_2(\mathbf{x})$	−1.4636	−1.4641	−1.4639	−1.4809	−1.4642	−1.4641075
$g_3(\mathbf{x})$	−0.5348	−0.5359	−0.5361	−0.5225	−0.5358	−0.535892
$f(\mathbf{x})$	263.9	263.8958434	263.8958466	264.3	263.895844071	**263.895843684**

DAPSO-GA: dynamic adaptive particle swarm optimization and genetic algorithm; DV: design variable; PSO-TVAC: Particle swarm optimization with time-varying accelerating coefficients.
Note: The boldfaced data in each table mean the best one among all the results provided by different algorithms.

Table 4. Comparison of statistical results obtained from different optimization algorithms for the three-bar truss design problem.

Algorithms	*M*	*S*	Worst	Mean	Best	SD	NFEs
DSS-MDE	10	300	263.8958498	263.8958436	263.8958434	9.72E–7	15,000
SC	20	1000	263.96975	263.9033	263.8958466	1.26E–2	17,610
PSO-TVAC	20	300	263.948096212	263.903085482	263.895848599	1.27E–02	6000
DAPSO-GA	20	300	**263.947633138**	**263.902926027**	**263.895843684**	**8.30E–03**	**7131**

SD: standard deviation; DAPSO-GA: dynamic adaptive particle swarm optimization and genetic algorithm; NFE: number of function evaluation.
Note: The boldfaced data mean optimal results provided by the DAPSO-GA algorithm.

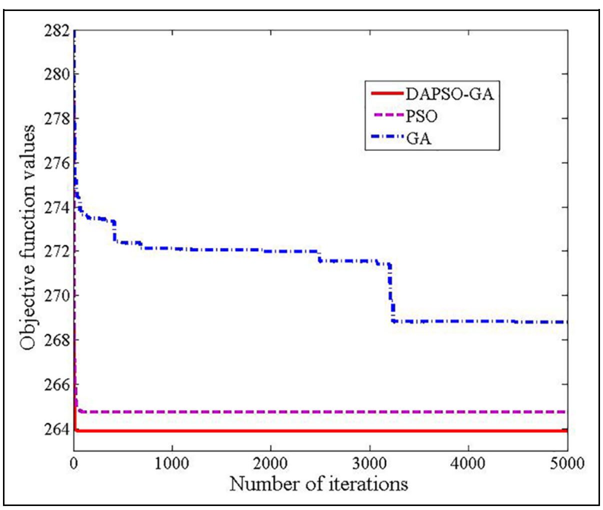

Figure 8. Convergence history of GA, standard PSO, and the proposed DAPSO-GA for the three-bar truss design problem.

of other optimization algorithms. DSS-MDE provides the best solution with the largest NFEs (15,000). Thus, the superiority of the proposed DAPSO-GA for the three-bar truss structure design problem in solution quality and convergence rate is justified. Figure 8 shows the convergence history of GA, standard PSO, and the proposed DAPSO-GA for the three-bar truss structure design problem. It is seen that the DAPSO-GA converges faster to the near optimal solution at early iterations and then gradually improves the solution accuracy due to the technique of the proposed algorithm in adaptively balancing the exploration and exploitation during searching process.

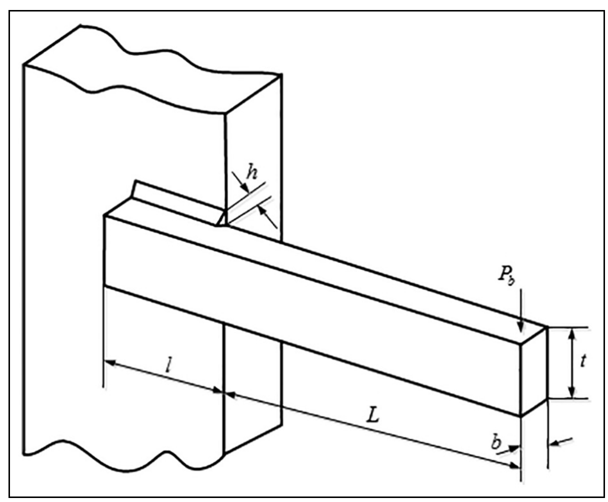

Figure 9. Schematic diagram of the welded beam.

Welded beam design. The welded beam design problem is a famous constrained optimization problem which is widely used as a benchmark problem to evaluate performance of newly proposed optimization algorithms.[9] Figure 9 shows the schematic diagram of a welded beam structure which consists of a beam and weld. The optimization target is the minimum fabrication cost of the beam subject to constraints on bending and shear stress (σ and τ) on the bar, bucking load (P_b), and its end deflections (δ). The design variables for this design problem are the weld thickness h, weld length l, beam width t, and beam thickness b, which are respectively denoted by x_1, x_2, x_3, and x_4 in the objective function and constraint equations as presented in Appendix 1

Table 5. Comparison of optimal solutions obtained from different optimization algorithms for the welded beam design optimization problem.

DV	GA3	IAPSO	APSO	MBA	WCA	NM-PSO	CAEP
x_1	0.2489	0.2057296	0.202701	0.205729	0.205728	0.205830	0.205700
x_2	0.1730	3.47048866	3.574272	3.470493	3.470522	3.468338	3.470500
x_3	8.1789	9.03662391	9.040209	9.036626	9.036620	9.036624	9.036600
x_4	0.2533	0.20572964	0.2059215	0.205729	0.205729	0.20573	0.205700
$g_1(\mathbf{x})$	−5758.604	−1.05E–10	−117.46706	−0.001614	−0.034128	−0.02525	1.988676
$g_2(\mathbf{x})$	−255.5769	−6.91E–10	−51.712981	−0.016911	−3.49E–05	−0.053122	4.481548
$g_3(\mathbf{x})$	−0.004400	−7.66E–15	−0.003221	−2.10E–7	−1.19E–06	0.000100	0
$g_4(\mathbf{x})$	−2.982866	−3.4329838	−3.421741	−3.432982	−3.432980	−3.433169	−3.433213
$g_5(\mathbf{x})$	−0.123900	−0.0807296	−0.077701	−0.080729	−0.080728	−0.080830	−0.080700
$g_6(\mathbf{x})$	−0.234160	−0.23554032	−0.235571	−0.235540	−0.235540	−0.235540	−0.235538
$g_7(\mathbf{x})$	−44.65.271	−5.80E–10	−18.367012	−0.001464	−0.013503	−0.031555	2.603347
$f(\mathbf{x})$	2.433116	1.7248523	1.736193	1.724853	1.724856	1.724717	1.724852

DV	CPSO	GA1	GA2	HPSO-GA	ABC2	DAPSO-GA
x_1	0.202369	0.2088	0.205986	0.2057296	0.2057245	0.205728318
x_2	3.544214	3.4205	3.471328	3.25312	3.25325369	2.994714573
x_3	9.048210	8.9975	9.020224	9.0366239	9.03664438	9.036612639
x_4	0.205723	0.2100	0.206480	0.2057296	0.20572999	0.205730191
$g_1(\mathbf{x})$	−13.655547	−0.337812	−0.103049	–	−0.17975428	−0.0732
$g_2(\mathbf{x})$	−78.814077	−353.9026	−0.231747	–	−0.18697948	−0.0057
$g_3(\mathbf{x})$	−3.35E–03	−0.0012	−5.0E–04	–	−0.00000549	−1.873E–6
$g_4(\mathbf{x})$	−3.424572	−3.411865	−3.430044	–	−3.45240767	−3.4755
$g_5(\mathbf{x})$	−0.077369	−0.0838	−0.080986	–	−0.08072450	−0.0807
$g_6(\mathbf{x})$	−0.235595	−0.235649	−0.235514	–	−0.22831066	−0.2355
$g_7(\mathbf{x})$	−4.472858	−363.2324	−58.64688	–	−0.03957707	−0.0434
$f(\mathbf{x})$	1.728024	1.7483094	1.78226	1.6952471	1.69526388	**1.6600473**

IAPSO: improved adaptive particle swarm optimization; APSO: accelerated particle swarm optimization; MBA: mine blast algorithm; CPSO: co-evolutionary particle swarm optimization; DAPSO-GA: dynamic adaptive particle swarm optimization and genetic algorithm; DV: design variables.
Note: The boldfaced data in each table mean the best one among all the results provided by different algorithms.

(section "Welded beam design"). The DAPSO-GA with a swarm size of 50 and maximum number of iterations of 5000 is used to solve this optimization problem.

The optimization algorithms previously used to solve this design optimization problem include GA3,[56] GA4,[35] APSO, IAPSO, MBA, LCA, WCA, DE, SC, NM-PSO, PSO-DE, HPSO,[29] CPSO,[24] CAEP, GA1, hybrid PSO-GA (HPSO),[39] ABC2,[40] and GA2. Table 5 presents the comparison of optimal solutions provided by the previously reported algorithms and proposed DAPSO-GA. From Table 5, a new optimal solution, which is better than those provided by previously proposed algorithms, is found by the proposed DAPSO-GA with the objective function value of 1.6600473. Note that the optimal solution provided by CAEP is infeasible as the constraints $g_1(\mathbf{x})$ and $g_2(\mathbf{x})$ are violated. Table 6 presents the comparison of statistical results provided by all previously reported algorithms and proposed DAPSO-GA for the welded beam design optimization problem in terms of the worst, mean, and best solutions as well as the SD and NFEs. As seen from Table 6, DAPSO-GA provides better solutions than the newly proposed optimization algorithm WCA, MBA, and IAPSO as well as other optimization algorithms. The proposed algorithm can stably find the best solution with almost the fewest NFEs (13,356) which is only larger than that of the IAPSO (12,500). In terms of SD, the proposed algorithm has better robustness in detecting the best solution than other reported optimization algorithms apart from the IAPSO, MBA, LCA, hybrid particle swarm optimization and genetic algorithm (HPSO-GA), ABC2, and PSO-DE. Figure 10 shows the convergence history of GA, standard PSO, and the proposed DAPSO-GA for the welded beam design problem. It is seen that the standard PSO and DAPSO-GA convergence faster than GA, while the DAPSO-GA has better global optimum searching ability.

Belleville disc spring design problem. As shown in Figure 11, Belleville disc spring is made up of several conical discs with uniform rectangular cross-sections. The design objective of the Belleville disc spring is to minimize its total weight subject to geometric constraints concerns the outer and inner diameter, slope and height to maximum height, and kinematic and strength constraints concerns the compression deformation and stress and height to deformation. There are four design variables for this

Table 6. Comparison of statistical results obtained from different optimization algorithms for the welded beam design optimization problem.

Algorithm	Worst	Mean	Best	SD	NFEs
GA3	2.64583	2.39203 (median)	2.38119	–	320,080
GA4	2.64583	2.39289 (median)	2.38119	–	40,080
APSO	1.993999	1.877851	1.736193	7.6118E–02	50,000
IAPSO	1.7248624	1.7248528	1.7248523	2.02E–06	12,500
MBA	1.724853	1.724853	1.724853	6.94E–19	47,340
LCA	1.7248523	1.7248523	1.7248523	7.11E–15	15,000
WCA	1.744697	1.726427	1.724856	4.29E–03	46,450
DE	1.824105	1.768158	1.733461	2.21E–02	204,800
SC	6.399678	3.002588	2.385434	9.60E–01	33,095
NM-PSO	1.733393	1.726373	1.724717	3.50E–03	80,000
PSO-DE	1.724852	1.724852	1.724852	6.70E–16	66,600
HPSO	1.814295	1.749040	1.724852	4.01E–02	81,000
CPSO	1.782143	1.748831	1.728024	1.29E–02	240,000
CAEP	3.179709	1.971809	1.724852	4.43E–01	50,020
GA1	1.785835	1.771973	1.748309	1.12E–02	900,000
GA2	1.993408	1.792654	1.728226	7.74E–02	80,000
HPSO-GA	1.6952741	1.6952741	1.6952741	2.192E–09	–
ABC2	1.6953706	1.6953084	1.69526388	2.84E–05	–
DAPSO-GA	**1.66876995**	**1.66043211083**	**1.66004730498**	**1.608237E–03**	**13,356**

SD: standard deviation; APSO: accelerated particle swarm optimization; IAPSO: improved adaptive particle swarm optimization; MBA: mine blast algorithm; CPSO: co-evolutionary particle swarm optimization; DAPSO-GA: dynamic adaptive particle swarm optimization and genetic algorithm.
Note: The boldfaced data mean optimal results provided by the DAPSO-GA algorithm.

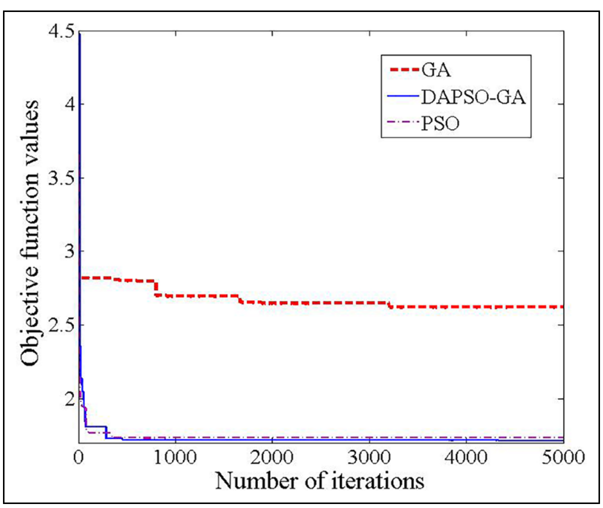

Figure 10. Convergence history of GA, standard PSO, and the proposed DAPSO-GA for the welded beam design problem.

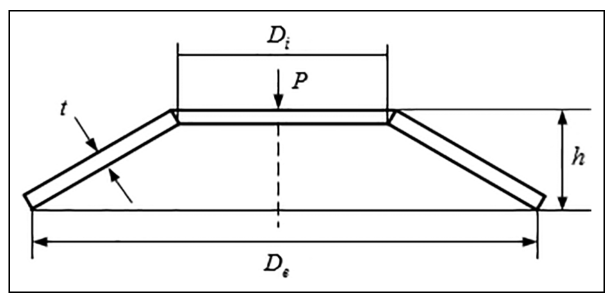

Figure 11. Schematic diagram of the Belleville disc spring.

design problem including the spring external and internal diameters (D_e and D_i), spring thickness (t), and spring height (h), which are denoted by x_1, x_2, x_3, and x_4, respectively. The DAPSO-GA with a swarm size of 50 and maximum number of iterations of 1000 is used to solve this optimization problem.

The optimization algorithms previously used to solve this design optimization algorithm include MBA, ABC, teaching-learning-based optimization (TLBO),[57] treating constrains as objectives (TCO),[58] Siddall,[59] Gene AS1,[60] and Gene AS2.[60] Table 7 presents the comparison of optimal solutions provided by the previously reported algorithms and proposed DAPSO-GA. Note that the optimal solutions provided by the Gene AS1 and Siddall are infeasible as the first and second constraints are violated by them, respectively. Hence, they are not used for comparison. From Table 7, the proposed algorithm and MBA provide better solutions against other optimization algorithms with the objective function value of 1.9796747. Table 8 presents the comparison of statistical results provided by the previously reported algorithms and proposed DAPSO-GA for the Belleville disc spring design optimization problem in terms of the worst, mean, and best solutions as well as the SD values and NFEs. As seen from Table 8, the proposed DAPSO-GA, ABC, TLBO, and MBA almost provide the same best solutions, but the proposed algorithm requires the fewest NFEs 9000 and ABC and TLBO requires the most NFEs 150,000. In terms of SD, MBA has better robustness in detecting the best solution than other optimization algorithms. Figure 12

Table 7. Comparison of optimal solutions obtained from different optimization algorithms for the Belleville disc spring design optimization problem.

DV	Coello	Gene AS1	Gene AS2	Siddall	TLBO	MBA	DAPSO-GA
x_1	0.208	0.205	0.210	0.204	0.204143	0.204143	0.20414335
x_2	0.2	0.201	0.204	0.200	0.20	0.20	0.2
x_3	8.751	9.534	9.268	10.03	10.03047	10.0304732	10.03047329
x_4	11.067	11.627	11.499	12.01	12.01	12.01	12.01
$g_1(\mathbf{x})$	2145.4109	−10.3396	2127.2624	134.0816	1.77E–06	4.58E–04	2.9296E–06
$g_2(\mathbf{x})$	39.75018	2.8062	194.222554	−12.5328	7.46E–08	3.04E–07	−6.7998E–08
$g_3(\mathbf{x})$	0	0.0010	0.0040	0	5.80E–11	9.24E–10	0.7797037
$g_4(\mathbf{x})$	1.592	1.5940	1.5860	1.596	1.595857	1.595856	1.59585664
$g_5(\mathbf{x})$	0.943	0.3830	0.5110	0	2.35E–09	0	0
$g_6(\mathbf{x})$	2.316	2.0930	2.2310	1.98	1.979527	1.979526	1.97952679
$g_7(\mathbf{x})$	0.21364	0.20397	0.20856	0.19899	0.198966	0.198965	0.1989657
$f(\mathbf{x})$	2.121964	2.01807	2.16256	1.978715	1.979675	1.9796747	**1.979674757**

MBA: mine blast algorithm; DAPSO-GA: dynamic adaptive particle swarm optimization and genetic algorithm.
Note: The boldfaced data in each table mean the best one among all the results provided by different algorithms.

Table 8. Comparison of statistical results obtained from different optimization algorithms for the Belleville disc spring design optimization problem.

Algorithm	Worst	Mean	Best	SD	NFEs
ABC	2.104297	1.995475	1.979675	0.07	150,000
TLBO	1.979757	1.979687	1.979675	0.45	150,000
MBA	2.005431	1.984698	1.9796747	7.78e–03	10,600
DAPSO-GA	**2.558209**	**2.132861**	**1.9796747**	**0.2358**	**9000**

SD: standard deviation; ABC: artificial bee colony; MBA: mine blast algorithm; DAPSO-GA: dynamic adaptive particle swarm optimization and genetic algorithm.
Note: The boldfaced data mean optimal results provided by the DAPSO-GA algorithm.

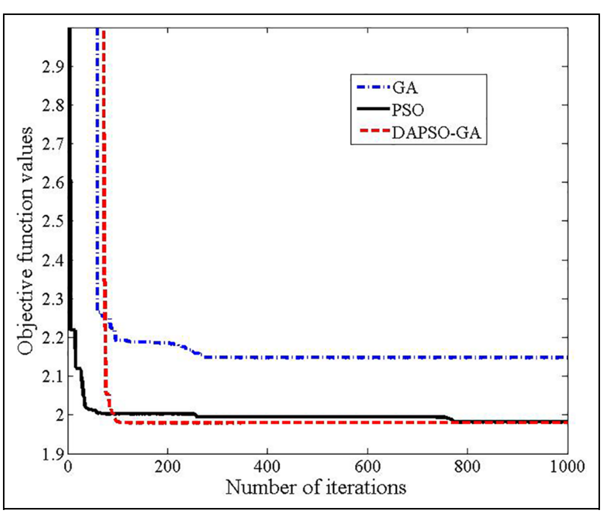

Figure 12. Convergence history of GA, standard PSO, and the proposed DAPSO-GA for the Belleville disc spring design problem.

shows the convergence history of GA, standard PSO, and the proposed DAPSO-GA for the Belleville disc design problem. It is seen that the standard PSO and DAPSO-GA convergence faster than GA, while the DAPSO-GA has better global optimum searching ability.

Constrained engineering optimization problems with discrete variables

Speed reducer design problem. Figure 13 shows a schematic diagram of a speed reducer. The design optimization scheme of the speed reducer is to minimize its weight subject to strength constraints concerning gear teeth bending stress and surface stress, stresses in and transverse deflections of shafts.[1] The design variables of this design problem include the face width (b), teeth module (m), number of teeth in the pinion (z), length of the first and second shafts between their bearings (l_1 and l_2), diameter of the first shaft (d_1), and diameter of the second shaft (d_2). These design variables are denoted by x_1, x_2, x_3, x_4, x_5, x_6, and x_7, respectively, in the objective function and constraint equations as presented in Appendix 2 (section "Speed reducer design problem"). The design variable x_3 (i.e. number of teeth in the pinion) is a discrete (integer) design variable and

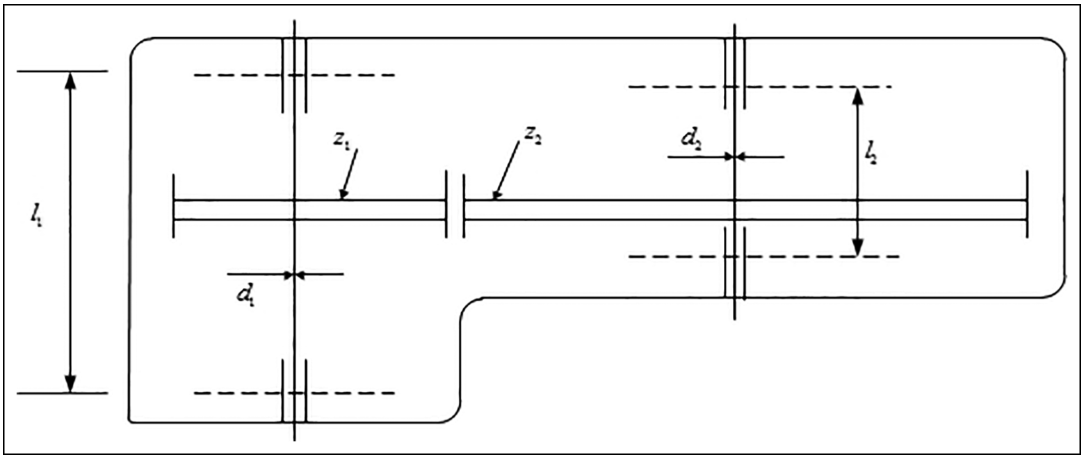

Figure 13. Schematic diagram of speed reducer.

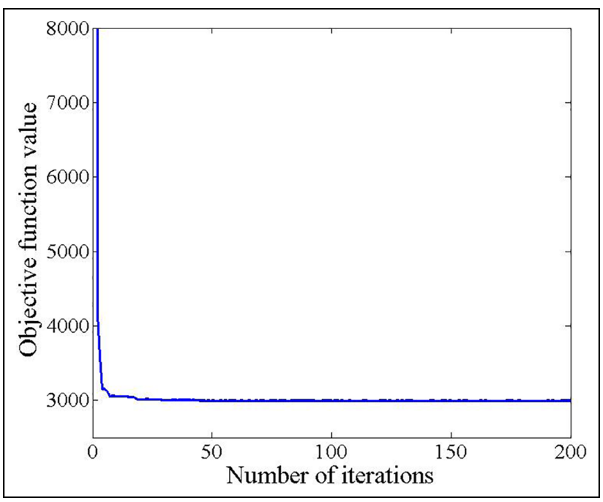

Figure 14. Convergence history of the proposed DAPSO-GA for the speed reducer design problem.

the remainder variables are continuous. The discrete DAPSO-GA with a swarm size of 30 and maximum number of iterations of 200 is used to solve this optimization problem. Figure 14 shows the convergence history of the proposed DAPSO-GA for the speed reducer design problem. The objective function value reduces fast to the near minimum at early iterations (less than 50 iterations), which presents the high convergence rate of this algorithm for this constrained discrete design problem.

This design optimization algorithm was previously solved by researchers using different optimization algorithms such as DEDS, DELC,[45] HEAA, MDE,[61] PSO-DE,[54] WCA, MBA, LCA, APSO, IAPSO, TLBO, ($\mu + \lambda$)-ES, SC, and ABC. Table 9 presents the comparison of optimal solutions provided by the previously reported algorithms and proposed DAPSO-GA. As seen from Table 9, the proposed algorithm and most of the reported algorithms including DEDS, DELC, HEAA, WCA, LCA, and IAPSO provide similar best solutions ($\mathbf{x}$ = (3.5, 0.7, 17, 7.3, 7.715319, 3.350214, 5.286654)) with the objective function value of 2994.4711. Table 10 presents the comparison of statistical results provided by the previously reported algorithms and proposed DAPSO-GA for the speed reducer design optimization problem in terms of the worst, mean, and best solutions as well as the SD values and NFEs. As seen from Table 10, DAPSO-GA and IAPSO can stably find the same best solutions with fewer number of iterations and medium SD value at the same time compared with other algorithms. Although DELC, differential evolution with dynamic stochastic selection (DEDS), and LCA can locate the best solution with the lowest level of SD (10^{-12}) among all algorithms, they need much more NFEs (30,000, 30,000, and 24,000, respectively) than DAPSO-GA (7320) and IAPSO (6000).

Gear train design problem. Figure 15 shows a schematic diagram of a gear train which consists of four gears. The scheme of the gear train design optimization problem is to minimize the error between the obtained gear ratio and the required gear ratio of 1/6.39[62] subject to constraints only on the allowable ranges of design variables (side constraints), which are the number of teeth of the four gears. It is a discrete optimization problem as all design variables are integers. Numbers of teeth of gears A, B, D, and F (i.e. design variables) in Figure 15 are respectively denoted by x_1, x_2, x_3, and x_4 in the objective function as presented in Appendix 2 (section "Gear train design problem"). The discrete DAPSO-GA with a swarm size of 30 and maximum number of iterations of 100 is used to solve this optimization problem.

This design problem was solved before by many researchers using different optimization algorithms

Table 9. Comparison of optimal solutions obtained from different optimization algorithms for the speed reducer design optimization problem.

DV	DEDS	DELC	HEAA	MDE	PSO-DE	WCA
x_1	3.5	3.5	3.500022	3.50001	3.5	3.5
x_2	0.7	0.7	0.7	0.7	0.7	0.7
x_3	17	17	17.000012	17	17	17
x_4	7.3	7.3	7.300427	7.300156	7.3	7.3
x_5	7.715319	7.715319	7.715377	7.800027	7.8	7.715319
x_6	3.350214	3.350214	3.350230	3.350221	3.350214	3.350214
x_7	5.286654	5.286654	5.286663	5.286685	5.2866832	5.286654
$g_1(\mathbf{x})$	−0.0739153	−0.0739153	−0.0739218	−0.0739179	−0.0739153	−0.0739153
$g_2(\mathbf{x})$	−0.197999	−0.197999	−0.198005	−0.198001	−0.197999	−0.197999
$g_3(\mathbf{x})$	−0.499172	−0.499172	−0.499094	−0.499144	−0.499172	−0.499172
$g_4(\mathbf{x})$	−0.904644	−0.904644	−0.904642	−0.901471	−0.901472	−0.904644
$g_5(\mathbf{x})$	5.9647E-07	5.9647E-07	−1.3025E-05	−5.4109E-06	5.9647E-07	5.9647E-07
$g_6(\mathbf{x})$	2.6369E-07	2.6369E-07	−4.8334E-06	2.2119E-08	1.6887E-08	2.6369E-07
$g_7(\mathbf{x})$	−0.7025	−0.7025	−0.7025	−0.7025	−0.7025	−0.7025
$g_8(\mathbf{x})$	0	0	−6.2857E-06	2.8571E-06	0	0
$g_9(\mathbf{x})$	−0.583333	−0.583333	−0.583331	−0.583332	−0.583333	−0.583333
$g_{10}(\mathbf{x})$	−0.0513259	−0.0513259	−0.0513781	−0.0513447	−0.051326	−0.0513259
$g_{11}(\mathbf{x})$	5.1845E-08	5.1845E-08	−6.1825E-06	−0.0108558	−0.0108524	5.1845E-08
$f(\mathbf{x})$	2994.4711	2994.4711	2994.49911	2996.35669	2996.34817	2994.47107

DV	MBA	LCA	APSO	IAPSO	DAPSO-GA
x_1	3.5	3.5	3.501313	3.5	3.5
x_2	0.7	0.7	0.7	0.7	0.7
x_3	17	17	18	17	17
x_4	7.300033	7.3	8.127814	7.3	7.3
x_5	7.715772	7.8	8.042121	7.71532	7.71531911
x_6	3.350218	3.350215	3.352446	3.3502147	3.35021467
x_7	5.286654	5.286683	5.287076	5.286654	5.28665447
$g_1(\mathbf{x})$	−0.0739153	−0.073915	−0.125692	−0.073915	−0.073915
$g_2(\mathbf{x})$	−0.197999	−0.197999	−0.284903	−0.197999	−0.198
$g_3(\mathbf{x})$	−0.499167	−0.499172	−0.34888	−0.499172	−0.499172
$g_4(\mathbf{x})$	−0.904627	−0.901472	−0.898038	−0.904644	−0.904644
$g_5(\mathbf{x})$	−2.9302E-06	4.0079E-13	−1.3515E-03	4.0079E-13	−1.4093E-10
$g_6(\mathbf{x})$	3.5053E-07	−4.785E-14	−2.6199E-04	2.6680E-11	−3.9881E-11
$g_7(\mathbf{x})$	−0.7025	−0.7025	−0.685	−0.7025	−0.7025
$g_8(\mathbf{x})$	0	0	−3.75E-04	−2.744E-11	−2.744E-11
$g_9(\mathbf{x})$	−0.583333	−0.583333	−0.583177	−0.583333	−0.583333
$g_{10}(\mathbf{x})$	−0.0513294	−0.051326	−0.147536	−0.051326	−0.0513258
$g_{11}(\mathbf{x})$	−5.8659E-06	−0.010852	−0.0405785	1.5542E-09	−5.04E-12
$f(\mathbf{x})$	2994.48245	2994.4711	3187.63049	2994.4711	**2994.47107**

MBA: mine blast algorithm; APSO: accelerated particle swarm optimization; IAPSO: improved adaptive particle swarm optimization; DAPSO-GA: dynamic adaptive particle swarm optimization and genetic algorithm; MDE: modified differential evolution.

such as Gene AS1, Gene AS2, SC, ABC, MBA, augmented Lagrangian (AL) method,[62] branch and bound (BB) method,[63] APSO, IAPSO, and UPSO.Table 11 presents the comparison of optimal solutions provided by the previously reported algorithms and proposed DAPSO-GA. According to the research of H Barbosa (September 1996, personal communication, San Francisco, CA) who computes all possible gear teeth combinations (49^4 or about 5.76 million), it can be validated that the optimal solutions provided by Gene AS1, ABC, and the proposed DAPSO-GA are globally best solutions. Whereas SC, MBA, APSO, and IAPSO find a different best solution as shown in Table 11. Statistical results provided by the previously reported algorithms and proposed DAPSO-GA for this design optimization problem are compared in terms of the worst, mean, and best solutions as well as the SD values and NFEs, as shown in Table 12. It is demonstrated that the proposed DAPSO-GA, MBA, and IAPSO are superior to other algorithms in terms of both SD and NFEs. The mean, best, and worst solutions provided by these three algorithms are at a same level, and they stably convergence to the best solution with similar computing efforts and SD values. Figure 16 shows the convergence history of the proposed DAPSO-GA for the gear train design problem.

Table 10. Comparison of statistical results obtained from different optimization algorithms for the speed reducer design optimization problem.

Algorithm	Worst	Mean	Best	SD	NFEs
SC	300.964736	3001.758264	2994.744241	4.0	54,456
PSO-DE	2996.348204	2996.348174	2996.348167	6.4E–06	54,350
DELC	2994.471066	2994.471066	2994.471066	1.9E–12	30,000
DEDS	2994.471066	2994.471066	2994.471066	3.6E–12	30,000
HEAA	2994.752311	2994.613368	2994.499107	7.0E–02	40,000
MDE	–	2996.367220	2996.356689	8.2E–03	24,000
$(\mu + \lambda)$-ES	–	2996.348000	2996.348000	0	30,000
ABC	–	2997.05800	2997.05800	0	30,000
TLBO	–	2996.34817	2996.34817	0	10,000
WCA	2994.505578	2994.474392	2994.471066	7.4E–03	15,150
LCA	2994.47106614683	2994.47106614682	2994.47106614682	2.66E–12	24,000
MBA	2999.652444	2996.769019	2994.482453	1.56	6300
APSO	4443.017639	3822.640624	3187.630486	366.146	30,000
IAPSO	2994.47106615489	2994.47106614777	2994.47106614598	2.65E–09	6000
DAPSO-GA	**2994.4713663**	**2994.4710726**	**2994.47106616**	**1.61415E–05**	**7320**

SD: standard deviation; ABC: artificial bee colony; MBA: mine blast algorithm; APSO: accelerated particle swarm optimization; IAPSO: improved adaptive particle swarm optimization; DAPSO-GA: dynamic adaptive particle swarm optimization and genetic algorithm.
Note: The boldfaced data mean optimal results provided by the DAPSO-GA algorithm.

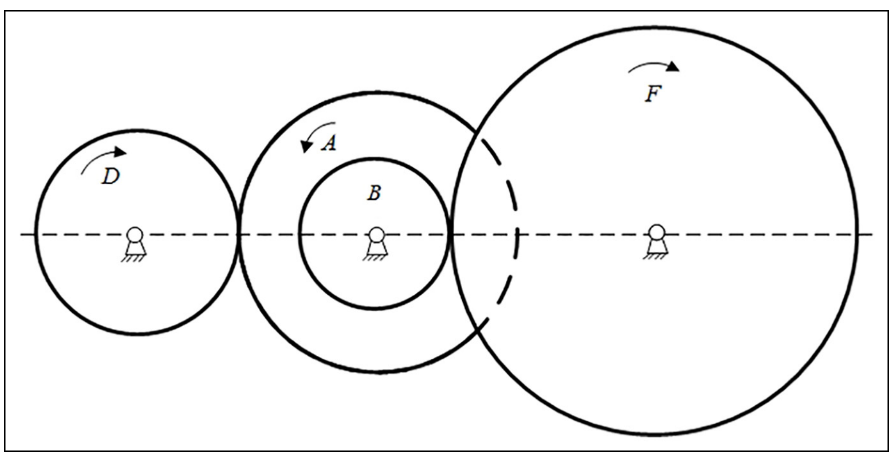

Figure 15. Schematic diagram of gear train.

Table 11. Comparison of optimal solutions obtained from different optimization algorithms for the gear train design optimization problem.

DV	Gene AS1	Gene AS2	SC	ABC	MBA	AL	BB	APSO	IAPSO	DAPSO-GA
x_1	49	33	43	49	43	33	45	43	43	49
x_2	16	14	16	16	16	15	22	16	16	16
x_3	19	17	19	19	19	13	18	19	19	19
x_4	43	50	49	43	49	41	60	49	49	43
$f(\mathbf{x})$	2.7 E–12	1.4E–09	2.7E–12	2.7E–12	2.7E–12	2.1E–08	5.7E–06	2.7E–12	2.7E–12	**2.7E–12**

ABC: artificial bee colony; MBA: mine blast algorithm; AL: augmented Lagrangian; BB: branch and bound; APSO: accelerated particle swarm optimization; IAPSO: improved adaptive particle swarm optimization; DAPSO-GA: dynamic adaptive particle swarm optimization and genetic algorithm.
Note: The boldfaced data in each table mean the best one among all the results provided by different algorithms.

Multiple disc clutch brake design problem. Figure 17 shows a schematic diagram of a multiple disc clutch brake. The design problem of the multiple disc clutch brake is a minimum problem which aims to minimize its total mass subject to geometrical constraints and constraints concerning shear stress, temperature, relative speed of

Table 12. Comparison of statistical results obtained from different optimization algorithms for the gear train design optimization problem.

Algorithm	Worst	Mean	Best	SD	NFEs
UPSO	–	3.80562E-08	2.700857E–12	1.09E–07	100,000
MBA	2.062904E–08	2.471635E–09	2.700857E–12	3.94E–09	1120
SC	2.3576E–09	1.9841E–09	2.7009E–12	3.5546E–09	5000
APSO	7.072678E–06	4.781676E–07	2.700857E–12	1.44E–06	8000
IAPSO	1.82738E–08	5.492477E–09	2.700857E–12	6.36E–09	800
DAPSO-GA	**2.7264505E–8**	**5.7898764E–09**	**2.70085714E–12**	**8.0549E–09**	**1438**

SD: standard deviation; MBA: mine blast algorithm; APSO: accelerated particle swarm optimization; IAPSO: improved adaptive particle swarm optimization; DAPSO-GA: dynamic adaptive particle swarm optimization and genetic algorithm.
Note: The boldfaced data in each table mean the best one among all the results provided by different algorithms.

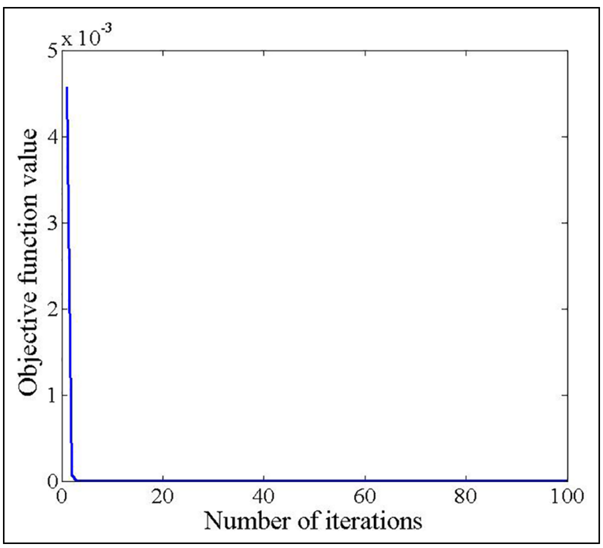

Figure 16. Convergence history of the proposed DAPSO-GA for the gear train design problem.

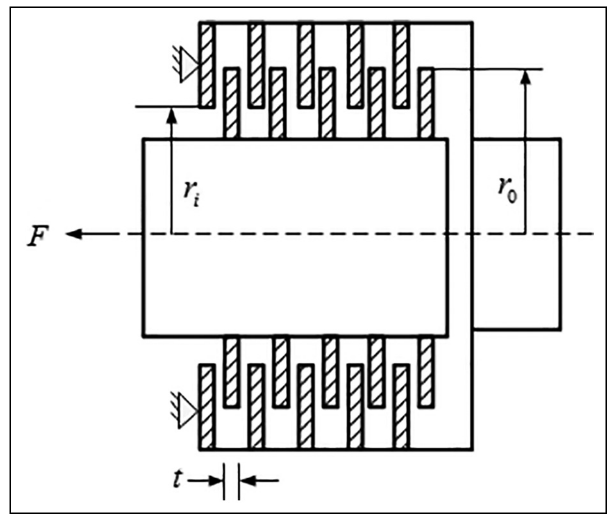

Figure 17. Schematic diagram of the multiple disc clutch brake.

the slip–stick, and stopping time.[64] The design variables for this design problem are inner and outer radius (r_i and r_0), disc thickness (A), actuating force (F), and number of contact surfaces (Z), which are denoted by x_1, x_2, x_3, x_4, and x_5, respectively. The objective variable x_4 only contains in the constraint equations (a side constraint). All design variables are discrete and should be selected from $x_1 = 60, 61, \ldots, 80$; $x_2 = 90, 91, \ldots, 110$; $x_3 = 1, 1.5, \ldots, 3$; $x_4 = 600, 610, \ldots, 1000$; $x_5 = 2, 3, \ldots, 9$. The discrete DAPSO-GA with a swarm size of 40 and maximum number of iterations of 100 is used to solve this optimization problem. All design variables are regarded as continuous variables and rounded off until at the end of the iterations. Besides, novel techniques are applied on the discrete variables x_3 and x_4 in this algorithm: x_3 is regarded as a continuous variable limited to the range [2, 6] and divided by two after being rounded to an integer; x_4 is regarded as a continuous variable limited to the range [60, 100] and multiplied by 10 after being rounded to an integer.

This design optimization problem was previously studied by many researchers using different optimization algorithms such as non-dominated sorting genetic algorithm (NSGA-II),[65] TLBO, WCA, ABC, APSO, and IAPSO. Table 13 presents the comparison of optimal solutions provided by the earlier reported algorithms and proposed DAPSO-GA. It is shown that the DAPSO-GA, IAPSO, WCA, and TLBO have the same objective function value of 0.31365661, although the values of the variable x_4 in the optimal solutions provided by these four algorithms are different. This is because x_4 only needs to satisfy the constraint conditions and is independent of the objective function. Statistical results provided by the previously reported algorithms and DAPSO-GA for this design optimization problem are compared as shown in Table 14. The statistical results demonstrate the superiority of the proposed DAPSO-GA against all proposed optimization algorithms in both NFEs and SD value. APSO performs the worst among all algorithms in terms of solution quality (mean and best solutions), SD value

Table 13. Comparison of optimal solutions obtained from different optimization algorithms for the multiple disc clutch brake design optimization problem.

DV	NSGA-II	TLBO	WCA	APSO	IAPSO	DAPSO-GA
x_1	70	70	70	76	70	70
x_2	90	90	90	96	90	90
x_3	1.5	1.0	1.0	1.0	1.0	1.0
x_4	1000	810	910	840	900	1000
x_5	3	3	3	3	3	3
$g_1(\mathbf{x})$	0	0	0	0	0	0
$g_2(\mathbf{x})$	22.00	24.00	24.00	24.00	24.00	24.00
$g_3(\mathbf{x})$	0.90052816	0.91942781	0.90948063	0.92227317	0.91047534	0.90052816
$g_4(\mathbf{x})$	9790.5816	9830.3711	9809.4293	9824.2113	9811.5234	9790.5816
$g_5(\mathbf{x})$	7894.6966	7894.6966	7894.6966	7738.378	7894.6966	7894.6966
$g_6(\mathbf{x})$	60,625.0	37,706.25	49,768.75	48,848.372	48,562.5	60,625.0
$g_7(\mathbf{x})$	11,647.293	14,297.987	12,768.578	12,873.649	12,906.636	11,647.293
$g_8(\mathbf{x})$	3352.7067	702.0132	2231.4215	2126.3515	2093.3635	3352.7067
$f(\mathbf{x})$	0.4704	0.313656	0.313656	0.337181	0.31365661	**0.31365661**

APSO: accelerated particle swarm optimization; IAPSO: improved adaptive particle swarm optimization; DAPSO-GA: dynamic adaptive particle swarm optimization and genetic algorithm.
Note: The boldfaced data in each table mean the best one among all the results provided by different algorithms.

Table 14. Comparison of statistical results obtained from different optimization algorithms for the multiple disc clutch brake design optimization problem.

Algorithm	Worst	Mean	Best	SD	NFEs
ABC	0.352864	0.324751	0.313657	–	>900
TLBO	0.392071	0.327166	0.313657	–	>900
WCA	0.313656	0.313656	0.313656	1.69E–16	500
APSO	0.716313	0.506829	0.337181	9.767E–02	2000
IAPSO	0.313656	0.313656	0.313656	1.13E–16	400
DAPSO-GA	**0.313656**	**0.313656**	**0.313656**	**1.129E–16**	**216**

SD: standard deviation; ABC: artificial bee colony; APSO: accelerated particle swarm optimization; IAPSO: improved adaptive particle swarm optimization; DAPSO-GA: dynamic adaptive particle swarm optimization and genetic algorithm.
Note: The boldfaced data mean optimal results provided by the DAPSO-GA algorithm.

and NFEs, and IAPSO and WCA ranks the second and third in terms of both SD value and NFEs, respectively.Figure 18 shows the convergence history of the proposed DAPSO-GA for the multiple disc clutch brake design problem. It quickly convergences to the best solution with less than 10 iterations due to the well balance between exploration and exploitation in searching process.

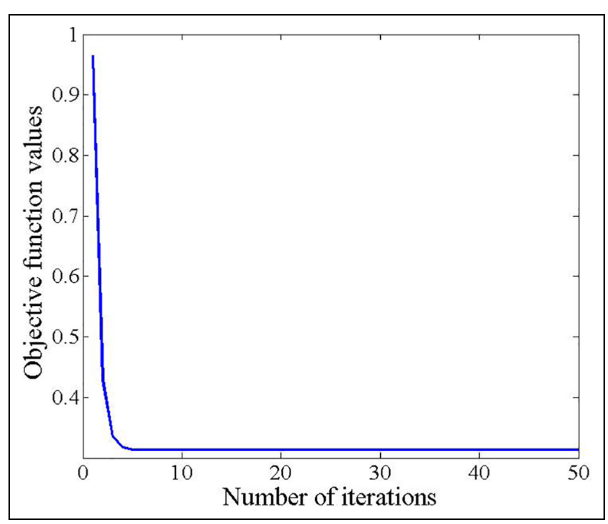

Figure 18. Convergence history of the proposed DAPSO-GA for the multiple disc clutch brake design problem.

Pressure vessel design problem. Figure 19 presents a schematic diagram of a pressure vessel. Two hemispherical heads are capped at the two ends of the cylindrical vessel. The pressure vessel design problem is first presented by Kannan and Kramer[62] and the design objective is to minimize its total fabricating cost including materials, forming, and welding costs. The design variables include the shell thickness T_s, head thickness T_h, inner radius R, and cylindrical section length of the vessel L,

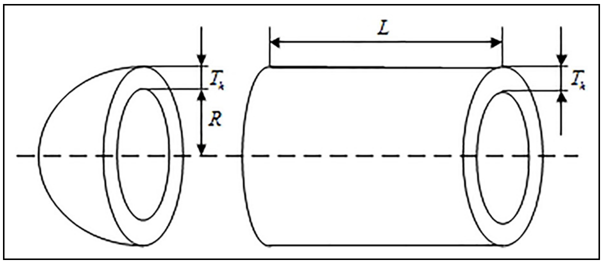

Figure 19. Schematic diagram of the pressure vessel.

in which T_s and T_h are discrete variables and are integer multiples of 0.0625 while R and L are continuous variables. These four design variables are respectively denoted by x_1, x_2, x_3, and x_4 in the objective function and constraint equations as presented in Appendix 2 (section "Pressure vessel design problem"). The proposed discrete DAPSO-GA with a swarm size of 25 and maximum number of iterations of 500 is used to solve this optimization problem. The discrete variables x_1 and x_2 are always kept as continuous variables limited in the range [0.5, 99.5]. Until at the end of the optimization process, x_1 and x_2 are not rounded to be integers and multiplied by 0.0625.

The pressure vessel design problem was previously studied by many researchers using different optimization algorithms including GA1, GA2, Cultural Differential Evolution (CDE),[66] PSO, CPSO, APSO, IAPSO, MBA, NM-PSO, G-QPSO, HPSO, WCA, HPSO-GA, ABC2, and LCA. The optimal solution obtained from the proposed algorithm is compared with those provided by the earlier reported algorithms as listed in Table 15. Table 16 presents the comparison of statistical results provided by the previously reported algorithms and proposed DAPSO-GA for the pressure vessel design optimization problem in terms of the worst, mean, and best solutions as well as the SD values and NFEs. It must be pointed out that the optimal results provided by NM-PSO, WCA MBA, HPSO-GA, and ABC are infeasible as the values of x_1 and x_2 are not integer multiples of 0.0625. Hence, only the remainder of the earlier algorithms listed in Tables 15 and 16 are compared with the proposed algorithm. From Tables 15 and 16, the proposed algorithm and IAPSO provide better solutions compared with other algorithms. Both of these two methods find the best solution with similar computation efforts (NFEs) which are fewer than those of other optimization algorithms, but IAPSO is more robust as its SD value is smaller. Figure 20 shows the convergence history of the proposed DAPSO-GA for the pressure vessel design problem.

Rolling element bearing design problem. The schematic diagram of a rolling element bearing is shown in Figure 21. The aim of the rolling element bearing design optimization is to maximize its dynamic loading bearing capacity subject to the geometric and kinematic constraints as well as the limit on the number of balls.[67] The design variables of this design optimization problem have five geometric parameters including the pitch diameter (D_m), ball diameter (D_b), number of balls (Z), inner and outer raceway curvature coefficients (f_i and f_0), and five other parameters only contain in the constraint equations ($K_{D\min}$, $K_{D\max}$, ε, e, and ξ). All design variables are continuous variables apart from the number of balls (Z). These 10 design variables are respectively denoted by x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, and x_{10}, respectively. The proposed discrete DAPSO-GA and PSO-TVAC with a swarm size of 50 and maximum number of iterations of 200 are used to solve this optimization problem.

Table 15. Comparison of optimal solutions obtained from different optimization algorithms for the pressure vessel design optimization problem.

DV	x_1	x_2	x_3	x_4	$g_1(\mathbf{x})$	$g_2(\mathbf{x})$	$g_3(\mathbf{x})$	$g_4(\mathbf{x})$	$f(\mathbf{x})$
GA1	0.8125	0.4375	42.0974	176.6540	−2.01E–03	−3.58E–02	−24.7593	−63.3460	6059.9463
GA2	0.8125	0.4375	42.0974	176.6540	−0.2E–05	−3.589E–02	−27.8861	−63.3460	6059.9463
CDE	0.8125	0.4375	42.0974	176.6376	−6.67E–07	−3.58E–02	−3.71051	−63.3623	6059.734
APSO	0.8125	0.4375	42.0974	176.6374	−9.54E–07	−3.59E–02	−63.3626	−0.9111	6059.7242
IAPSO	0.8125	0.4375	42.0974	176.6366	−4.09E–13	−3.58E–02	−1.39E–07	−63.3634	6059.7143
CPSO	0.8125	0.4375	42.0913	176.7465	−1.37E–06	−3.59E–04	−118.7687	−63.2535	6061.0777
MBA	0.7802	0.3856	40.4292	198.4694	0	0	−86.3645	−41.5035	5889.3216
NM-PSO	0.8036	0.3972	41.6392	182.412	3.65E–05	3.79E–05	−1.5914	−57.5879	5930.3137
G-QPSO	0.8125	0.4375	42.0984	176.6372	−8.79E–07	−3.58E–02	−0.2179	−63.3628	6059.7208
WCA	0.7781	0.3846	40.3196	200.0000	−2.95E–11	−7.15E–11	−1.35E–6	−40.00	5885.3327
HPSO-GA	0.7782	0.3846	40.3196	200.0000	0	0	−4.656E–10	−40	5885.3328
ABC2	0.7782	0.3847	0.3211	199.9802	−1.40E–06	−2.84E–06	−1.1418	−40.0197	5885.4033
DAPSO-GA	0.8125	0.4375	42.0984	176.6366	−4.09E–13	−3.58E–02	−1.39E–07	−63.3634	**6059.7143**

APSO: accelerated particle swarm optimization; IAPSO: improved adaptive particle swarm optimization; CPSO: co-evolutionary particle swarm optimization; MBA: mine blast algorithm; DAPSO-GA: dynamic adaptive particle swarm optimization and genetic algorithm.
Note: The boldfaced data in each table mean the best one among all the results provided by different algorithms.

Table 16. Comparison of statistical results obtained from different optimization algorithms for the pressure vessel design optimization problem.

Algorithm	Worst	Mean	Best	SD	NFEs
PSO	14,076.324	8756.6803	6693.7212	1492.5670	8000
APSO	7544.49272	6470.71568	6059.7242	326.9688	200,000
IAPSO	6090.5314	6068.7539	6059.7143	14.0057	7500
MBA	6392.5062	6200.64765	5889.3216	160.34	70,650
LCA	6090.6114	6070.5884	6059.8553	11.37534	24,000
WCA	6590.2129	6198.6172	5885.3327	213.0490	27,500
CDE	6371.0455	6085.2303	6059.7340	43.0130	204,800
GA1	6308.4970	6293.8432	6288.7445	7.4133	900,000
GA2	6469.3220	6177.2533	6059.9463	130.9297	80,000
QPSO	8017.2816	6440.3786	6059.7209	479.2671	8000
G-QPSO	7544.4925	6440.3786	6059.7208	448.4711	8000
NM-PSO	5960.0557	5946.7901	5930.3137	9.1610	80,000
HPSO	6288.6770	6099.9323	6059.7143	86.2000	81,000
CPSO	6363.8041	6147.1332	6061.0777	86.4500	240,000
HPSO-GA	5885.4864	5885.3821	5885.3328	0.049	
ABC2	5895.1268	5887.5570	5885.4032	2.7453	–
DAPSO-GA	**7319.0007**	**6267.1671**	**6059.7143**	**380.9406**	**9000**

SD: standard deviation; PSO: particle swarm optimization; APSO: accelerated particle swarm optimization; IAPSO: improved adaptive particle swarm optimization; MBA: mine blast algorithm; CPSO: co-evolutionary particle swarm optimization; DAPSO-GA: dynamic adaptive particle swarm optimization and genetic algorithm.
Note: The boldfaced data mean optimal results provided by the DAPSO-GA algorithm.

This design optimization problem was previously solved by many researchers using different optimization algorithms such as GA5,[67] ABC, TLBO, and MBA. Optimal solutions given by these reported algorithms and PSO-TVAC are compared with those provided by the proposed DAPSO-GA in terms of the values of design variables, objective function value, and constraint accuracy, as detailed in Table 17. It must be emphasized that there are some errors for the optimal solutions of GA5, TLBO, and MBA given by Sadollah et al.[9] in terms of the objective function value, number of constraints, and constraint accuracy, which are revised in this work as shown in Table 17. Note that the optimal solutions provided by GA5 and TLBO are infeasible as the fourth constraint $g_4(\mathbf{x})$ is violated. Hence, their optimal solutions are not used for later comparison. Table 18 presents the comparison of statistical results provided by the previously reported algorithms and proposed DAPSO-GA for the rolling element bearing design optimization problem in terms of the worst, mean, and best solutions as well as the SD values and NFEs. As seen from Table 18, the proposed algorithm finds the best solution (81,859.80912) with the fewest NFEs (3650). ABC stably provides the similar best solution (81,859.7416) with the smallest SD value but much more NFEs than the proposed algorithm. PSO-TVAC convergences to the similar best solution (81,859.7415974) with similar NFEs (3750) but much larger SD value compared with the proposed algorithm. Figure 22 shows the convergence history of the proposed DAPSO-GA for the rolling element bearing design problem. Note that the proposed DAPSO-GA convergences fast to the best solution with less than 50 iterations thanks to the global optima searching technique.

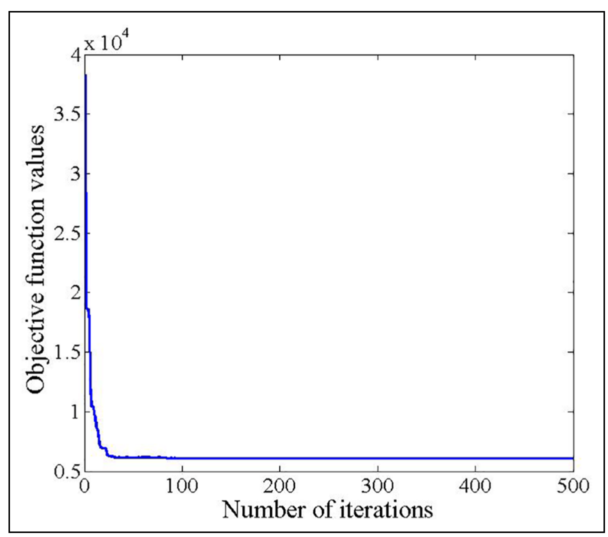

Figure 20. Convergence history of the proposed DAPSO-GA for the pressure vessel design problem.

Conclusion

In this work, a DAPSO-GA is presented to solve constrained engineering design optimization problems with

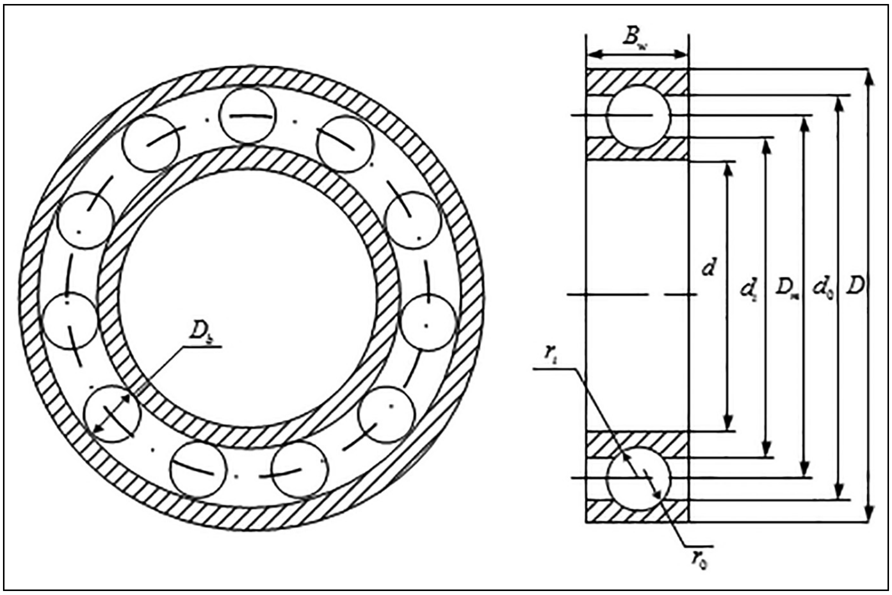

Figure 21. Schematic diagram of rolling element bearing.

Table 17. Comparison of optimal solutions obtained from different optimization algorithms for the rolling element bearing design optimization problem.

DV	GA5	TLBO	MBA	PSO-TVAC	DAPSO-GA
x_1	125.7171	125.7191	125.7153	125.7191	125.7191
x_2	21.423	21.4259	21.4233	21.4256	21.4256
x_3	11	11	11	11	11
x_4	0.515	0.515	0.515	0.5150	0.515
x_5	0.515	0.515	0.515	0.5150	0.515
x_6	0.4159	0.424266	0.488805	0.4169	0.4000
x_7	0.651	0.633948	0.627829	0.7000	0.7000
x_8	0.300043	0.3	0.300149	0.3000	0.3000
x_9	0.0223	0.068858	0.097305	0.1000	0.0474
x_{10}	0.751	0.799498	0.64095	0.6001	0.6000
$g_1(\mathbf{x})$	8.22E–04	−1.2235E–07	5.6382E–04	3.0198E–014	−1.2235E–07
$g_2(\mathbf{x})$	13.732999	13.15257	8.63025	14.8511804817	14.8512
$g_3(\mathbf{x})$	2.724000	1.52518	1.10143	3.64614410764	6.1488
$g_4(\mathbf{x})$	1.107	2.559363	−2.04045	−3.42559024083	−3.4256
$g_5(\mathbf{x})$	0.717100	4.7191	0.7153	0.719055614672	0.7191
$g_6(\mathbf{x})$	4.857899	16.49544	23.61095	24.2809443853	11.1309
$g_7(\mathbf{x})$	0.0021288	−2.999E–05	5.179283E–04	1.4797E–012	−2.999E–05
$g_8(\mathbf{x})$	0	0	0	0	0
$g_9(\mathbf{x})$	0	0	0	0	0
$f(\mathbf{x})$	81,841.5108	81,859.74	81,843.68625	81,859.74159741332	**81,859.80912**

MBA: mine blast algorithm; DAPSO-GA: dynamic adaptive particle swarm optimization and genetic algorithm.
Note: The boldfaced data in each table mean the best one among all the results provided by different algorithms.

different kinds of objective functions, design variables, and constraints in nature. The presented algorithm uses a dynamic adaptive inertia weighting factor, which adaptively adjusts the search velocity in optimum searching process, to balance the exploitation (local search) and exploration (global search). In the proposed algorithm, GA-related operators are incorporated into PSO and used to refine the optimal solution provided by the PSO. Few particles in the swarm that meet the GA-selection criterion with time-varying selection probability are adaptively selected to update their positions via a crossover and n-point mutation operator in each iteration process. Global best and worst positions of the PSO are updated according to the refined particle position generated by GA. With the three GA-related operators, the particle swarm is greatly diversified and as a result, premature convergence is effectively prevented. The promising prospect of the proposed

Table 18. Comparison of statistical results obtained from different optimization algorithms for the rolling element bearing design optimization problem.

Algorithm	Worst	Mean	Best	SD	NFEs
GA5	–	–	81,843.3	–	225,000
ABC	78,897.810	81,496.00	81,859.7416	0.69	10,000
TLBO	80,807.8551	81,438.987	81,859.7400	0.66	10,000
MBA	–	–	81,843.68625	211.52	15,100
PSO-TVAC	41,130.5723727	76,442.495342	81,859.7415974	11,888.8903	3750
DAPSO-GA	**79,834.7818**	**81,066.4318**	**81,859.80912**	**742.9211**	**3650**

SD: standard deviation; ABC: artificial bee colony; MBA: mine blast algorithm; DAPSO-GA: dynamic adaptive particle swarm optimization and genetic algorithm.
Note: The boldfaced data mean optimal results provided by the DAPSO-GA algorithm.

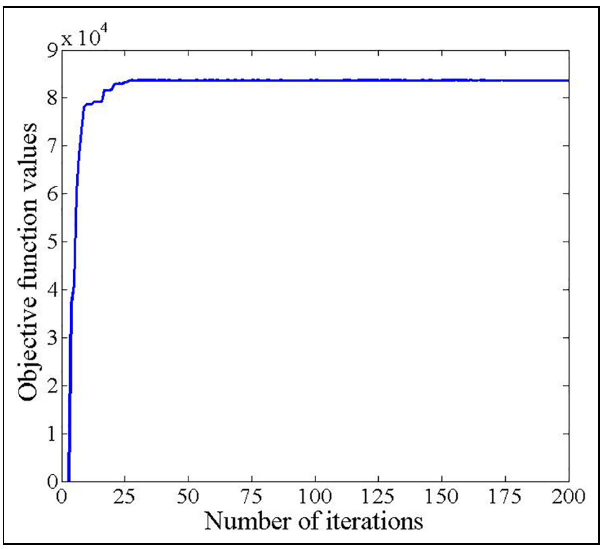

Figure 22. Convergence history of the proposed DAPSO-GA for the rolling element bearing design problem.

DAPSO-GA for engineering constrained optimization problems is evaluated by solving nine different benchmark mechanical engineering design optimization problems with continuous, discrete, or mixed design variables. For most of the considered mechanical engineering design optimization problems, statistical results show that the proposed DAPSO-GA convergences to the best or similar solution with the smallest SD values and lowest computation efforts (NFEs) against other meta-heuristic algorithms.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: The authors gratefully acknowledge to the financial support from the National Natural Science Foundation of China (no. 51805339), Fundamental Research Funds for Central Universities and the State Key Development Program for Basic Research of China (no. 2014CB049401).

ORCID iDs

Hao Zhu https://orcid.org/0000-0003-3910-2947
Weidong Zhu https://orcid.org/0000-0003-2707-2533

References

1. Guedria NB. Improved accelerated PSO algorithm for mechanical engineering optimization problems. *Appl Soft Comput* 2016; 40: 455–467.
2. Kennedy J and Eberhart R. Particle swarm optimization. In: *IEEE international conference on neural networks (ICNN)*, Perth, WA, Australia, vol. 4, 27 November–1 December 1995, pp.1942–1948. New York: IEEE.
3. Holland JH. Outline for a logical theory of adaptive systems. *J ACM* 1962; 9: 297–314.
4. Bagley JD. *The behavior of adaptive systems which employ genetic and correlation algorithms*. Ann Arbor, MI: Dissertation Abstracts International, University of Michigan, 1967.
5. Goldberg DE. *Genetic algorithms in search, optimization and machine learning*. Boston, MA: Addison-Wesley Longman Publishing Co., Inc., 1989.
6. Yang XS. Firefly algorithms for multimodal optimization. In: Watanabe O and Zeugmann T (eds) *Stochastic algorithms: foundations and applications*, vol. 5792. Berlin: Springer, 2009, pp.169–178.
7. Dorigo M, Birattari M and Stutzle T. Ant colony optimization artificial ants as a computational intelligence technique. *IEEE Comput Intell M* 2006; 1: 28–39.
8. Karaboga D and Basturk B. Artificial bee colony (ABC) optimization algorithm for solving constrained optimization. In: Melin P, Castillo O, Aguilar LT, et al. (eds) *Foundations of fuzzy logic and soft computing*. *IFSA 2007* (Lecture Notes in Computer Science, vol. 4529). Berlin: Springer, 2007, pp.789–798.
9. Sadollah A, Bahreininejad A, Eskandar H, et al. Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems. *Appl Soft Comput* 2013; 13: 2592–2612.

10. Kirkpatrick S, Gelatt CD and Vecchi MP. Optimization by simulated annealing. *Science* 1983; 220: 671–680.
11. Garg H. An efficient biogeography based optimization algorithm for solving reliability optimization problems. *Swarm Evol Comput* 2015; 24: 1–10.
12. Gouttefarde M and Gosselin CM. Analysis of the wrench-closure workspace of planar parallel cable-driven mechanisms. *IEEE T Robot* 2006; 22: 434–445.
13. Perez RE and Behdinan K. Particle swarm optimization in structural design. In: Chan FTS and Tiwari MK (eds) *Swarm intelligence: focus on ant and particle swarm optimization*. Vienna: Intech Education and Publishing, 2007, pp.373–394.
14. Hassan R, Cohanim BE and de Weck OL. A comparison of particle swarm optimization and the genetic algorithm. In: *46th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics, and materials conference*, Austin, TX, 18–21 April 2005, AIAA paper no. 2005–1897. Reston, VA: American Institute of Aeronautics and Astronautics.
15. Mortazavi A and Toğan V. Simultaneous size, shape, and topology optimization of truss structures using integrated particle swarm optimizer. *Struct Multidiscip O* 2016; 54: 715–736.
16. Joshua TB, Xin J and Sunil KA. Optimal design of cable-driven manipulators using particle swarm optimization. *J Vib Acoust* 2016; 8: 041003.
17. Esmin AAA and Matwin S. HPSOM: a hybrid particle swarm optimization algorithm with genetic mutation. *Int J Innov Comput I* 2013; 9: 1919–1934.
18. Lovbjerg M and Krink T. Extending particle swarm optimizers with self-organized critically. In: *Proceedings of the 2002 congress on evolutionary computation. CEC'02*, vol. 2, Honolulu, HI, 12–17 May 2002, pp.1588–1593. New York: IEEE.
19. Shi Y and Eberhart R. Empirical study of particle swarm optimization. In: *Proceedings of the 1999 congress on evolutionary computation-CEC99*, Washington, DC, 6–9 July 1999, pp.1945–1950. New York: IEEE.
20. Eberhart RC and Shi Y. Tracking and optimizing dynamic systems with particle swarms. In: *Proceedings of the 2001 congress on evolutionary computation*, Seoul, Korea, 27–30 May 2001, pp.94–97. New York: IEEE.
21. Clerc M. The Swarm and The Queen: towards a deterministic and adaptive particle swarm optimization. In: *Proceedings of the 1999 congress on evolutionary computation*, Washington, DC, 6–9 July 1999. New York: IEEE.
22. Ratnaweera A, Halgamuge SK and Harry CW. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE T Evolut Comput* 2004; 8: 240–255.
23. Shi Y and Eberhart RC. Parameter selection in particle swarm optimization. In: Porto VW, Saravanan N, Waagen D, et al. (eds) *Evolutionary programming VII*. Berlin: Springer, 1998, pp.591–600.
24. He Q and Wang L. An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Eng Appl Artif Intel* 2007; 20: 89–99.
25. Krohling RA and Coelho LDS. Coevolutionary particle swarm optimization using Gaussian distribution for solving constrained optimization problems. *IEEE T Syst Man Cy B* 2006; 36: 1407–1416.
26. Worasucheep C. Solving constrained engineering optimization problems by the constrained PSO-DD. In: *2008 5th international conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, Krabi, Thailand, 14–17 May 2008, pp.5–8. New York: IEEE.
27. Yang XS. *Engineering optimization: an introduction with metaheuristic applications*. Hoboken, NJ: John Wiley & Sons, Inc., 2010.
28. Yang XS, Deb S and Fong S. Accelerated particle swarm optimization and support vector machine for business optimization and applications. In: Fong S (ed.) *Networked digital technologies. NDT 2011. Communications in computer and information science*, vol. 136. Berlin: Springer, 2011, pp.53–66.
29. Novitasari D, Cholissodin I and Mahmudy WF. Hybridizing PSO with SA for optimizing SVR applied to software effort estimation. *TELKOMNIKA* 2016; 14: 245–253.
30. He Q and Wang L. A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. *Appl Math Comput* 2007; 186: 1407–1422.
31. Wang J and Yin Z. A ranking selection-based particle swarm optimizer for engineering design optimization problems. *Struct Multidiscip O* 2008; 37: 131–147.
32. Lei J-J and Li J. A modified particle swarm optimization for practical engineering optimization. In: *2009 fifth international conference on natural computation*, vol. 3, Tianjin, China, 14–16 August 2009, pp.177–180. New York: IEEE.
33. Coelho LDS. Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems. *Expert Syst Appl* 2010; 37: 1676–1683.
34. Mahmoodabadi MJ and Bisheban M. An online optimal linear state feedback controller based on MLS approximations and a novel straightforward PSO algorithm. *T I Meas Control* 2014; 36: 1132–1142.
35. Zhu H, Hu YM, Zhu WD, et al. Optimal design of an auto-tensioner in an automotive belt drive system via a dynamic adaptive PSO-GA. *J Mech Design* 2017; 139: 093302.
36. Zhang M, Luo W and Wang X. Differential evolution with dynamic stochastic selection for constrained optimization. *Inform Sciences* 2008; 178: 3043–3074.
37. Coello CAC. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Comput Method Appl M* 2002; 191: 1245–1287.
38. Mezura-Montes E and Coello CAC. A simple multimembered evolution strategy to solve constrained optimization problems. *IEEE T Evolut Comput* 2005; 9: 1–17.
39. Garg H. A hybrid PSO-GA algorithm for constrained optimization problems. *Appl Math Comput* 2016; 274: 292–305.
40. Garg H. Solving structural engineering design optimization problems using an artificial bee colony algorithm. *J Ind Manag Optim* 2014; 10: 777–794.
41. Coello CAC. Use of a self-adaptive penalty approach for engineering optimization problems. *Comput Ind* 2000; 41: 113–127.

42. Coello CAC and Montes EM. Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Adv Eng Inform* 2002; 16: 193–203.
43. Eskandar H, Sadollah A, Bahreininejad A, et al. Water cycle algorithm—a novel metaheuristic optimization method for solving constrained engineering optimization problems. *Comput Struct* 2012; 110: 151–166.
44. Lampinen J. A constraint handling approach for the differential evolution algorithm. In: *Proceedings of the 2002 congress on evolutionary computation. CEC'02*, Honolulu, HI, 12–17 May 2002, pp.1468–1473. New York: IEEE.
45. Wang L and Li LP. An effective differential evolution with level comparison for constrained engineering design. *Struct Multidiscip O* 2010; 41: 947–963.
46. Zahara E and Kao YT. Hybrid Nelder–Mead simplex search and particle swarm optimization for constrained engineering design problems. *Expert Syst Appl* 2009; 36: 3880–3886.
47. Wang Y, Cai Z, Zhou Y, et al. Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique. *Struct Multidiscip O* 2009; 37: 395–413.
48. Davoodi E, Hagh MT and Zadeh SG. A hybrid improved quantum-behaved particle swarm optimization-simplex method (IQPSOS) to solve power system load flow problems. *Appl Soft Comput* 2014; 21: 171–179.
49. Ray T and Liew KM. Society and civilization: an optimization algorithm based on the simulation of social behavior. *IEEE T Evolut Comput* 2003; 7: 386–396.
50. Kashan AH. An efficient algorithm for constrained global optimization and application to mechanical engineering design: league championship algorithm (LCA). *Comput Aided Design* 2011; 43: 1769–1792.
51. Parsopoulos KE and Vrahatis MN. Unified particle swarm optimization for solving constrained engineering optimization problems. In: Wang L, Chen K and Ong YS (eds) *Advances in natural computation*. Berlin: Springer, 2005, pp.582–591.
52. Mezura-Montes E and Coello CAC. Useful infeasible solutions in engineering optimization with evolutionary algorithms. In: Gelbukh A, de Albornoz Á and Terashima-Marín H (eds) *MICAI 2005: advances in artificial intelligence* (Lecture Notes in Computer Science, vol. 3789). Berlin: Springer, 2005, pp.652–662.
53. Liu H, Cai Z and Wang Y. Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. *Appl Soft Comput* 2010; 10: 629–640.
54. Henendz S. Multiobjective structural optimization. In: Kodoyalam S and Saxena M (eds) *Geometry and optimization techniques for structural design*. Amsterdam: Elsevier, 1994, pp.341–362.
55. Ray T and Saini P. Engineering design optimization using a swarm with an intelligent information sharing among individuals. *Eng Optimiz* 2001; 33: 735–748.
56. Deb K. Optimal design of a welded beam via genetic algorithms. *AIAA J* 1991; 29: 2013–2015.
57. Rao RV, Savsani VJ and Vakharia DP. Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. *Comput Aided Design* 2011; 43: 303–315.
58. Coello CAC. Treating constraints as objectives for single-objective evolutionary optimization. *Eng Optimiz* 2000; 32: 275–308.
59. Siddall JN. *Optimal engineering design: principles and applications*. New York: Marcel Dekker, 1982.
60. Deb K and Goyal M. Optimizing engineering designs using a combined genetic search. In: *Proceedings of the seventh international conference on genetic algorithms*, East Lansing, MI, 19–23 July 1997, pp.521–528. San Francisco, CA: Morgan Kaufmann Publishers Inc.
61. Mezura-Montes E, Velázquez-Reyes J and Coello CAC. Modified differential evolution for constrained optimization. In: *2006 IEEE international conference on evolutionary computation*, Vancouver, BC, Canada, 16–21 July 2006, pp.25–32. New York: IEEE.
62. Kannan BK and Kramer SN. An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. *J Mech Design* 1994; 116: 405–411.
63. Sandgren E. Nonlinear integer and discrete programming in mechanical design. *J mech Design* 1990; 112: 223–229.
64. Osyczka A. *Evolutionary algorithms for single and multicriteria design optimization* (Studies in Fuzziness and Soft Computing). Heidelberg: Physica-Verlag, 2002.
65. Deb K and Srinivasan A. Innovization: innovating design principles through optimization. In: *Proceedings of the 8th annual conference on genetic and evolutionary computation*, Seattle, WA, 8–12 July 2006, pp.1629–1636. New York: ACM Press.
66. Becerra RL, Coello CAC. Cultured differential evolution for constrained optimization. *Computer Methods in Applied Mechanics and Engineering* 2006; 195: 4303–4022.
67. Gupta S, Tiwari R and Nair SB. Multi-objective design optimization of rolling bearings using genetic algorithms. *Mech Mach Theory* 2007; 42: 1418–1443.

Appendix 1

Tension/compression spring design problem

$$\text{Minimize} \quad f(\mathbf{x}) = (x_3 + 2)x_2x_1^2$$

Subject to

$$g_1(\mathbf{x}) = 1 - \frac{x_2^3x_3}{71785x_1^4} \leqslant 0$$

$$g_2(\mathbf{x}) = \frac{4x_2^2 - x_1x_2}{12566\left(x_2x_1^3 - x_1^4\right)} + \frac{1}{5108x_1^2} - 1 \leqslant 0$$

$$g_3(\mathbf{x}) = 1 - \frac{140.45x_1}{x_2^2x_3} \leqslant 0$$

$$g_4(\mathbf{x}) = \frac{x_1 + x_2}{1.5} - 1 \leqslant 0$$

where $0.05 \leqslant x_1 \leqslant 20$, $0.25 \leqslant x_2 \leqslant 1.3$, and $2.00 \leqslant x_3 \leqslant 15.00$.

Symmetric three-bar truss design problem

$$
\begin{aligned}
&\text{Minimize} \quad f(\mathbf{x}) = (2\sqrt{2}x_1 + x_2)l \\
&\text{Subject to} \\
&g_1(\mathbf{x}) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leqslant 0,\ g_2(\mathbf{x}) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leqslant 0,\ g_3(\mathbf{x}) = \frac{\sqrt{2}}{\sqrt{2}x_1 + 2x_2}P - \sigma \leqslant 0
\end{aligned}
$$

where $0 \leqslant x_1 \leqslant 1$, $0 \leqslant x_2 \leqslant 1$, $l = 100\,\text{cm}$, $P = 2\,\text{KN/cm}^2$, and $\sigma = 2\,\text{KN/cm}^2$.

Welded beam design

$$
\begin{aligned}
&\text{Minimize} \quad f(\mathbf{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2) \\
&\text{Subject to} \\
&g_1(\mathbf{x}) = \tau(\mathbf{x}) - \tau_{\max} \leqslant 0,\ g_2(\mathbf{x}) = \sigma(\mathbf{x}) - \sigma_{\max} \leqslant 0,\ g_3(\mathbf{x}) = x_1 - x_4 \leqslant 0 \\
&g_4(\mathbf{x}) = 0.10471x_1^2 + 0.04811x_3x_4(14 + x_2) - 5 \leqslant 0,\ g_5(\mathbf{x}) = 0.125 - x_1 \leqslant 0 \\
&g_6(\mathbf{x}) = \delta(\mathbf{x}) - 0.25 \leqslant 0,\ g_7(\mathbf{x}) = P - P_c(\mathbf{x}) \leqslant 0
\end{aligned}
$$

where $0.1 \leqslant x_1,\ x_4 \leqslant 2$ and $0.1 \leqslant x_2,\ x_3 \leqslant 10$.

Belleville spring design problem

$$
\begin{aligned}
&\text{Minimize} \quad f(\mathbf{x}) = 0.07075\pi\left(D_e^2 - D_i^2\right)t \\
&\text{Subject to} \\
&g_1(\mathbf{x}) = S - \frac{4E\delta_{\max}}{(1-\mu^2)\alpha D_e^2}[\beta(h - \delta_{\max}/2) + \gamma t] \geqslant 0 \\
&g_2(\mathbf{x}) = \frac{4E\delta_{\max}}{(1-\mu^2)\alpha D_e^2}\left[(h - \delta_{\max}/2)(h - \delta_{\max})t + t^3\right] - P_{\max} \geqslant 0 \\
&g_3(\mathbf{x}) = \delta_l - \delta_{\max} \geqslant 0,\ g_4(\mathbf{x}) = H - h - t \geqslant 0,\ g_5(\mathbf{x}) = D_{\max} - D_e \geqslant 0 \\
&g_6(\mathbf{x}) = D_e - D_i \geqslant 0,\ g_7(\mathbf{x}) = 0.3 - \frac{h}{D_e - D_i} \geqslant 0
\end{aligned}
$$

where

$$
\begin{aligned}
\alpha &= \frac{6}{\pi \ln K}\left(\frac{K-1}{K}\right)^2 \\
\beta &= \frac{6}{\pi \ln K}\left(\frac{K-1}{\ln K} - 1\right) \\
\gamma &= \frac{6}{\pi \ln K}\left(\frac{K-1}{2}\right)
\end{aligned}
$$

$P_{\max} = 5400$ lb, $\delta_{\max} = 0.2$ in, $S = 200$ kpsi, $E = 30E06$ psi, $\mu = 0.3$, $H = 2$ in, $D_{\max} = 12.01$ in, $K = D_e/D_i$, $\delta_l = f(a)a$, $a = h/t$.

Values of vary as detailed in Table 19.

Table 19. Variation of $f(a)$ with a.

a	$\leqslant 1.4$	1.5	1.6	1.7	1.8	1.9	2.0	2.1	2.2	2.3	2.4	2.5	2.6	2.7	$\geqslant 2.8$
$f(a)$	1	0.85	0.77	0.71	0.66	0.63	0.60	0.58	0.56	0.55	0.53	0.52	0.51	0.51	0.50

Appendix 2

Speed reducer design problem

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

Subject to

$$g_1(\mathbf{x}) = \frac{27}{x_1x_2^2x_3} - 1 \leqslant 0,\ g_2(\mathbf{x}) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \leqslant 0,\ g_3(\mathbf{x}) = \frac{1.93x_4^3}{x_2x_6^4x_3} - 1 \leqslant 0,\ g_4(\mathbf{x}) = \frac{1.93x_5^3}{x_2x_7^4x_3} - 1 \leqslant 0$$

$$g_5(\mathbf{x}) = \frac{\left[(745x_4/x_2x_3)^2 + 16.9\times 10^6\right]^{0.5}}{110x_6^3} - 1 \leqslant 0,\ g_6(\mathbf{x}) = \frac{\left[(745x_5/x_2x_3)^2 + 157.5\times 10^6\right]^{0.5}}{85x_7^3} - 1 \leqslant 0$$

$$g_7(\mathbf{x}) = x_2x_3/40 - 1 \leqslant 0,\ g_8(\mathbf{x}) = \frac{5x_2}{x_1} - 1 \leqslant 0,\ g_9(\mathbf{x}) = \frac{x_1}{12x_2} - 1 \leqslant 0$$

$$g_{10}(\mathbf{x}) = \frac{1.5x_6 + 1.9}{x_4} - 1 \leqslant 0,\ g_{11}(\mathbf{x}) = \frac{1.1x_7 + 1.9}{x_5} - 1 \leqslant 0$$

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

Gear train design problem

$$\text{Minimize}\quad f(\mathbf{x}) = \left(\frac{1}{6.931} - \frac{x_2x_3}{x_1x_4}\right)^2$$

Subject to

$12 \leqslant x_i \leqslant 60,\ i = 1, 2, 3, 4$

Multiple disc clutch brake design problem

$$\text{Minimize}\quad f(\mathbf{x}) = \pi\left(r_0^2 - r_i^2\right)(Z + 1)\rho t$$

Subject to

$$g_1(\mathbf{x}) = r_0 - r_i - \Delta r \geqslant 0,\ g_2(\mathbf{x}) = l_{\max} - (Z + 1)(t + \delta) \geqslant 0,\ g_3(\mathbf{x}) = P_{\max} - P_{rz} \geqslant 0$$

$$g_4(\mathbf{x}) = P_{\max}v_{sr\max} - P_{rz}v_{sr} \geqslant 0,\ g_5(\mathbf{x}) = v_{sr\max} - v_{sr} \geqslant 0,\ g_6(\mathbf{x}) = T_{\max} - T \geqslant 0$$

$$g_7(\mathbf{x}) = M_h - sM_s \geqslant 0,\ g_8(\mathbf{x}) = T \geqslant 0$$

where

$$M_h = \frac{2}{3}\mu FZ\frac{r_0^3 - r_i^3}{r_0^2 - r_i^2}$$

$$P_{rz} = \frac{2}{3}\frac{F}{\pi\left(r_0^2 - r_i^2\right)}$$

$$v_{rz} = \frac{2\pi n\left(r_0^3 - r_i^3\right)}{90\left(r_0^2 - r_i^2\right)}$$

$$T = \frac{I_z\pi n}{30\left(M_h - M_f\right)}$$

$\Delta r = 20$ mm, $I_z = 55\,\text{kgm}^2$, $P_{\max} = 1$ MPa, $F_{\max} = 1000$ N, $T_{\max} = 15$ s, $\mu = 0.5$, $s = 1.5$, $M_s = 40$ N m, $M_f = 3$ N m, $n = 250$ r/min, $v_{sr\max} = 10$ m/s, $l_{\max} = 30$ mm, 60 mm $\leqslant r_i \leqslant$ 80 mm, 90 mm $\leqslant r_0 \leqslant$ 110 mm, 1.5 mm $\leqslant t \leqslant$ 3 mm, 600 N $\leqslant F \leqslant$ 1000 N, $2 \leqslant Z \leqslant 9$.

Pressure vessel design problem

$$\begin{aligned}
&\text{Minimize} \quad f(\mathbf{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.166x_1^2x_4 + 19.84x_1^2x_3\\
&\text{Subject to}\\
&g_1(\mathbf{x}) = -x_1 + 0.0193x_3 \leqslant 0\\
&g_2(\mathbf{x}) = -x_2 + 0.00954x_3 \leqslant 0\\
&g_3(\mathbf{x}) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leqslant 0\\
&g_4(\mathbf{x}) = x_4 - 240 \leqslant 0
\end{aligned}$$

where $1 \times 0.0625 \leqslant x_1,\ x_2 \leqslant 99 \times 0.0625$ and $10 \leqslant x_3,\ x_4 \leqslant 200$.

Rolling element bearing design problem

$$\begin{aligned}
&\text{Maximize} \quad f(\mathbf{x}) = C_d = \begin{cases} f_c Z^{2/3} D_b^{1.8}, & \textit{if } D_b \leqslant 25.4 \text{ mm}\\ 3.647 f_c Z^{2/3} D_b^{1.4}, & \textit{else} \end{cases}\\
&\text{Subject to}\\
&g_1(\mathbf{x}) = \frac{\phi_0}{2\sin^{-1}(D_b/D_m)} - Z + 1 \geqslant 0\\
&g_2(\mathbf{x}) = 2D_b - K_{D\min}(D-d) \geqslant 0\\
&g_3(\mathbf{x}) = K_{D\min}(D-d) - 2D_b \geqslant 0\\
&g_4(\mathbf{x}) = \xi B_w - D_b \geqslant 0\\
&g_5(\mathbf{x}) = D_m - 0.5(D+d) \geqslant 0\\
&g_6(\mathbf{x}) = (0.5+e)(D+d) - D_m \geqslant 0\\
&g_7(\mathbf{x}) = 0.5(D - D_m - D_b) - \varepsilon D_b \geqslant 0\\
&g_8(\mathbf{x}) = f_i - 0.515 \geqslant 0\\
&g_9(\mathbf{x}) = f_0 - 0.515 \geqslant 0
\end{aligned}$$

where

$$f_c = 37.91\left\{1 + \left[1.04\left(\frac{1-\gamma}{1+\gamma}\right)^{1.72}\left(\frac{f_i(2f_0-1)}{f_0(2f_i-1)}\right)^{0.41}\right]^{10/3}\right\}^{-0.3}\left[\frac{\gamma^{0.3}(1-\gamma)^{1.39}}{(1+\lambda)^{1/3}}\right]\left(\frac{2f_i}{2f_i-1}\right)^{0.41}$$

$$\phi_0 = 2\pi - 2\cos^{-1}\left(\frac{[(D-d)/2 - 3T/4]^2 + (D/2 - T/4 - D_b)^2 - (d/2 + T/4)^2}{2[(D-d)/2 - 3T/4](D/2 - T/4 - D_b)}\right)$$

$\gamma = D_b/D_m$, $f_i = r_i/D_b$, $f_0 = r_0/D_b$, $T = D - d - 2D_b$, $D = 160$, $d = 90$, $B_w = 30$, $0.4 \leqslant K_{D\min} \leqslant 0.5$, $0.6 \leqslant K_{D\max} \leqslant 0.7$, $0.3 \leqslant \varepsilon \leqslant 0.4$, $0.02 \leqslant e \leqslant 0.1$, $0.6 \leqslant \xi \leqslant 0.85$, $0.5(D+d) \leqslant D_m \leqslant 0.6(D+d)$, $0.15(D-d) \leqslant D_b \leqslant 0.45(D-d)$.