Chin. J. Mech. Eng. (2017) 30:553–565
DOI 10.1007/s10033-017-0130-4

ORIGINAL ARTICLE

Improved Differential Evolution with Shrinking Space Technique for Constrained Optimization

Chunming FU[1] · Yadong XU[2] · Chao JIANG[1] · Xu HAN[1] · Zhiliang HUANG[1]

Received: 4 May 2016 / Revised: 10 February 2017 / Accepted: 2 April 2017 / Published online: 24 April 2017

Abstract Most of the current evolutionary algorithms for constrained optimization algorithm are low computational efficiency. In order to improve efficiency, an improved differential evolution with shrinking space technique and adaptive trade-off model, named ATMDE, is proposed to solve constrained optimization problems. The proposed ATMDE algorithm employs an improved differential evolution as the search optimizer to generate new offspring individuals into evolutionary population. For the constraints, the adaptive trade-off model as one of the most important constraint-handling techniques is employed to select better individuals to retain into the next population, which could effectively handle multiple constraints. Then the shrinking space technique is designed to shrink the search region according to feedback information in order to improve computational efficiency without losing accuracy. The improved DE algorithm introduces three different mutant strategies to generate different offspring into evolutionary population. Moreover, a new mutant strategy called "DE/rand/best/1" is constructed to generate new individuals according to the feasibility proportion of current population. Finally, the effectiveness of the proposed method is verified by a suite of benchmark functions and practical engineering problems. This research presents a constrained evolutionary algorithm with high efficiency and accuracy for constrained optimization problems.

Keywords Constrained optimization · Differential evolution · Adaptive trade-off model · Shrinking space technique

Supported by National Science Foundation for Excellent Young Scholars, China (Grant No. 51222502), Funds for Distinguished Young Scientists of Hunan Province, China (Grant No. 14JJ1016), and Major Program of National Natural Science Foundation of China (Grant No. 51490662).

✉ Yadong XU
13601580051@139.com

1 State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, Hunan University, Changsha 410082, China

2 School of Mechanical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China

1 Introduction

Constrained optimization problems (COPs) widely exist in various scientific and engineering fields [1–3], such as mechanical design, path planning, etc. Perhaps it is not easy or difficult to obtain global optimal solutions by the traditional optimization techniques for some COPs involving nonlinear inequality or equality constraints, multi-modal and non-differential objective functions. Evolutionary algorithms (EAs) cooperated with constraint-handling techniques which have obtained more and more attention because of their flexibility, effectiveness and adaptability provide an effective and powerful avenue to cope with these COPs [4–6]. A large number of effective constrained optimization evolutionary algorithms (COEAs) have been proposed [7–9]. Recently, some representative constraint-handling techniques with EAs to solve COEAs have been summarized by COELLO [10]. The most general existing constraint-handling techniques are mainly categorized into three groups. Firstly, the method based on the penalty function aimed to transform a COP into an unconstrained one by adding a penalty term to the original objective function [11, 12]. Secondly, the approach based on the feasibility-based criterion preferred to select the feasible solutions rather than the infeasible solutions into the next evolutionary process [13, 14]. Thirdly, the method based on the multi-objective optimization technique aimed to transform the COPs into the unconstrained multi-objective optimization problems and utilized multi-objective optimization technique to deal with the converted problems [15, 16].

The performance of COEAs mainly depends on the constraint-handling techniques and EAs as the search optimizer. Differential evolution (DE) originally proposed by STORN and PRICE [17] was one of the most simple and powerful population-based evolutionary algorithms for global optimization. During the past two decades, different DE optimizers with constraint-handling techniques have been successfully developed to deal with different kinds of COPs. The first attempt was the constraint adaption with DE (CADE) algorithm which introduced multi-member individuals to generate more than one offspring by DE operators [18]. A cultural DE-based algorithm with the feasibility rule was proposed by LANDA and COELLO [19], which utilized different knowledge sources to influence the mutant operator in order to accelerate convergence. A multi-member diversity-based DE (MDDE) algorithm where each parent generated more than one offspring to enhance the diversity of population was presented by MEZURA-MONTES, et al [20] to solve COPs. The dynamic stochastic selection technique was put forward by ZHANG, et al [21] under the framework of multi-member DE. TESSEMA and YEN [11] designed an adaptive penalty formulation where the feasible proportion of the current population was utilized to tune the penalty factor. In order to combine the advantages of different constraint-handling techniques, MALLIPEDDI, et al [22] proposed an ensemble of constraint-handling techniques (ECHT) with DE and evolutionary programming optimizers for coping with COPs. ELSAYED, et al [23] introduced an algorithm framework to use multiple search operators in each generation with the feasibility rule for COPs. Each combination of search operators had its own sub-population, and the size of each sub-population varied adaptively during the progress of evolution depending on the reproductive success of the search operators. Subsequently, GONG, et al [24] developed a ranking-based mutation operator with an improved dynamic diversity mechanism for COPs. A modified differential evolution algorithm [25] was proposed to deal with the dimensional synthesis of the redundant parallel robot problem.

Recently, the adaptive trade-off model (ATM) [26] has been proposed to maintain a reasonable tradeoff to select better individuals to reserve into next generation between the feasible and infeasible individuals. The principal merit of ATM was that the promising infeasible individuals could be inherited into the next evolutionary process. The ATM with evolutionary strategy (ATMES) as the search optimizer has been utilized to solve COPs. In order to reduce the computational effort, the shrinking space technique introduced by AGUIRRE, et al [27] shrank the search region according to some feedback information and directed the search effort to the promising feasible region. Subsequently, WANG, et al [28] proposed a new method named AATM with high efficiency which benefited from the virtues of shrinking space technique and ATM. The performance of AATM algorithm could promptly converge to optimal results without loss of quality and precision.

Although AATM enhances the performance of ATMES by taking advantage of the shrinking space technique to address complicated COPs with multiple constraints, it still leaves a plenty of room to develop new approaches to solve COPs for improvement of accelerating the convergence rate and enhancing the quality of solutions within the limited time, especially for complicated engineering optimization problems. When using EAs to solve COPs, the search algorithm plays a crucial role on the performance of hybrid approaches as well as the constraint-handling techniques. Hence, this study employs an advanced search algorithm (i.e. an improved DE) to further improve the performance of AATM. The improved DE employs three different characteristic mutant strategies to generate different offspring into evolutionary population. Hence, combining the advantages of an improved differential evolution with adaptive trade-off model and shrinking space technique, called ATMDE, is proposed to deal with COPs. The remainders of this paper are organized as follows. In Section 2, the definitions of COP and some relevant concepts of multi-objective optimization are given, respectively. In Section 3, the basics of DE are briefly introduced. In Section 4, the proposed ATMDE algorithm is presented in detail. In Section 5, the performances of ATMDE are tested by 18 well-known benchmark test functions from the 2006 IEEE Congress on Evolutionary Computation (IEEE CEC2006) and several engineering optimization problems. Section 6 concludes this paper.

2 Statement of the Problem

A general constrained optimization problem is formulated as

$$\min f(\boldsymbol{x}),\\ \text{s.t.}\begin{cases} g_k(\boldsymbol{x}) \le 0, & k = 1, 2, \cdots, q, \\ h_k(\boldsymbol{x}) = 0, & k = q+1, q+2, \cdots, m, \end{cases} \tag{1}$$

where $f(\boldsymbol{x})$ is the objective function; $g(\boldsymbol{x})$ and $h(\boldsymbol{x})$ are the inequality and equality constraints, respectively; $\boldsymbol{x} = [x_1, x_2, \cdots, x_n]$ is an n-dimensional vector of design variables and their allowable lower and upper boundaries are $x_{\min,j}$ and $x_{\max,j}$ ($j = 1, 2,\cdots, n$); m is the total number of constraints; q and $m - q$ are the numbers of inequality and equality constraints, respectively. In evolutionary optimization, equality constraints are transformed to inequality constraints as follows:

$$|h_k(\boldsymbol{x})| - \delta \leq 0, \tag{2}$$

where δ is a positive tolerance parameter and is recommended to be as 0.0001 [20, 21].

In addition, the degree of violation value of solution $\boldsymbol{x}$ from k-th constraint $G_k(\boldsymbol{x})$ is defined as

$$G_k(\boldsymbol{x}) = \begin{cases} \max\{0, g_k(\boldsymbol{x})\}, & k = 1, 2, \cdots, q, \\ \max\{0, |h_k(\boldsymbol{x})| - \delta\}, & k = q+1, q+2, \cdots, m. \end{cases} \tag{3}$$

Then, the degree of all the constraint violations of solution $\boldsymbol{x}$ can be represented as

$$G(\boldsymbol{x}) = \sum_{k=1}^{m} G_k(\boldsymbol{x}). \tag{4}$$

Since the following method utilizes the concepts of the multi-objective optimization techniques to address constraints of COPs, some related multi-objective optimization concepts are introduced in advance.

Definition 1 *Pareto dominance*: A vector $\boldsymbol{u} = (u_1, u_2,\cdots, u_k)$ is said to *Pareto dominate* another vector $\boldsymbol{v} = (v_1, v_2,\cdots, v_k)$, denoted as $\boldsymbol{u} \prec \boldsymbol{v}$, only if it is satisfied:

$$\forall i \in \{1, 2, \cdots, k\},\ u_i \leq v_i \quad \text{and} \quad \exists j \in \{1, 2, \cdots, k\},\ u_j < v_j. \tag{5}$$

Definition 2 *Pareto optimality*: $\boldsymbol{u}$ is said to be *Pareto optimal* only if vector $\boldsymbol{v}$ in the feasible region S doesn't exist and $\boldsymbol{v} \prec \boldsymbol{u}$, where $\boldsymbol{v} = \boldsymbol{f}(\boldsymbol{v}) = (f(\boldsymbol{v}), G(\boldsymbol{v})) = (v_1, v_2)$, $\boldsymbol{u} = \boldsymbol{f}(\boldsymbol{u}) = (f(\boldsymbol{u}), G(\boldsymbol{u})) = (u_1, u_2)$.

Definition 3 *Pareto optimal set*: The Pareto optimal set denoted as P^* is defined as

$$P^* = \{\boldsymbol{u} \in S | \neg \exists \boldsymbol{v} \in S, \boldsymbol{v} \prec \boldsymbol{u}\}. \tag{6}$$

It should be noted that individuals in the Pareto optimal set are called *non-dominated* individuals.

Definition 4 *Pareto front*: The Pareto front PF^* is defined as

$$PF^* = \{\boldsymbol{f}(\boldsymbol{u}) | \boldsymbol{u} \in P^*\}. \tag{7}$$

3 Basics of Differential Evolution

DE has been extensively applied to solve optimization problems because of its simplicity and effectiveness [17]. It does not require the binary encoding to represent solution like genetic algorithm and not employ a probability density function to self-adapt its individuals like evolution strategy. It generates new candidate solutions by combining the parent individual and several other individuals of the current population. Then a candidate individual will replace the parent only if it has better fitness value. In the following text, the specific operations including initialization, mutation, crossover and selection are introduced. Firstly, it generates NP initial population $\boldsymbol{x}_i$ ($i = 1, 2,\cdots, NP$) sampled from the search domain by

$$x_{i,j} = x_{\min,j} + \text{rand}(0, 1) \times (x_{\max,j} - x_{\min,j}), \tag{8}$$

where rand(0,1) means to generate a randomly real number between 0 and 1.

After initialization, a mutant strategy is adopted to generate a mutant vector $\boldsymbol{v}_i = (v_{i,1}, v_{i,2},\cdots, v_{i,n})$ by its corresponding target vector $\boldsymbol{x}_i = (x_{i,1}, x_{i,2},\cdots, x_{i,n})$. There is a general nomenclature "DE/x/y" developed to denote the different DE mutant variants, where "DE" means differential evolution, "x" indicates which individual as the base vector is selected to be mutated, and "y" is the number of difference vectors chosen for perturbation of x. The following mutation strategies are most frequently used.

DE/rand/1:

$$\boldsymbol{v}_i = \boldsymbol{x}_{r_1} + F \times (\boldsymbol{x}_{r_2} - \boldsymbol{x}_{r_3}). \tag{9}$$

DE/best/1:

$$\boldsymbol{v}_i = \boldsymbol{x}_{\text{best}} + F \times (\boldsymbol{x}_{r_1} - \boldsymbol{x}_{r_2}). \tag{10}$$

DE/rand/2:

$$\boldsymbol{v}_i = \boldsymbol{x}_{r_1} + F \times (\boldsymbol{x}_{r_2} - \boldsymbol{x}_{r_3}) + F \times (\boldsymbol{x}_{r_4} - \boldsymbol{x}_{r_5}). \tag{11}$$

DE/current-to-rand/1:

$$\boldsymbol{v}_i = \boldsymbol{x}_i + F \times (\boldsymbol{x}_{r_1} - \boldsymbol{x}_i) + F \times (\boldsymbol{x}_{r_2} - \boldsymbol{x}_{r_3}). \tag{12}$$

DE/current-to-best/1:

$$\boldsymbol{v}_i = \boldsymbol{x}_i + F \times (\boldsymbol{x}_{\text{best}} - \boldsymbol{x}_i) + F \times (\boldsymbol{x}_{r_1} - \boldsymbol{x}_{r_2}). \tag{13}$$

where indices r_1, r_2, r_3, r_4 and r_5 are mutually exclusive integers randomly selected from interval [1, NP] and are also different from individual i; F is the scale factor chosen between 0 and 1; and $\boldsymbol{x}_{\text{best}}$ denotes the best individual in the current population.

Subsequently, a trial vector $\boldsymbol{u}_i = (u_{i,1}, u_{i,2},\cdots, u_{i,n})$ generates by the binomial crossover or exponential crossover. The binomial crossover is utilized in this paper as follows:

$$u_{i,j} = \begin{cases} v_{i,j}, & \text{if } (\text{rand}_j(0, 1) \leq CR) \text{ } or \text{ } j = j_{\text{rand}}, \\ x_{i,j}, & \text{otherwise}. \end{cases} \tag{14}$$

where CR is the probability rate of crossover operator and j_{rand} is a randomly integer chosen from the range [1, n]. The binomial crossover operator inherits the j-th variable of mutant vector $\boldsymbol{v}_i$ to its corresponding element in the trial vector $\boldsymbol{u}_i$ if it meets the condition. Taking "DE/rand/1/bin"

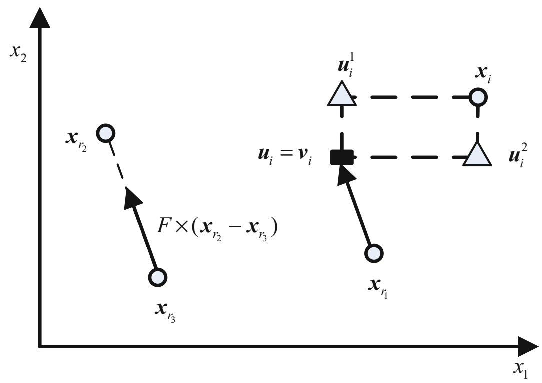

Fig. 1 Schematic diagram of "DE/rand/1/bin" strategy (2-D space)

strategy as an example, the schematic diagram of mutation and crossover operation is illustrated as shown Fig. 1. The black square represents the mutant vector, which is the mutant vector $\boldsymbol{u}_i$ generated by mutant strategy. The two triangles $\boldsymbol{u}_i^1$ and $\boldsymbol{u}_i^2$ represent the two possible locations for the trial vector after performing binomial crossover operation.

Then, the target vector $\boldsymbol{x}_{i,g}$ compares with its trial vector $\boldsymbol{u}_{i,g}$ by their fitness values, and the better one $\boldsymbol{x}_{i,g+1}$ will survive into the next generation population. The selection operation expresses as

$$\boldsymbol{x}_{i,g+1} = \begin{cases} \boldsymbol{u}_{i,g}, & \text{if } f(\boldsymbol{u}_{i,g}) \leq f(\boldsymbol{x}_{i,g}), \\ \boldsymbol{x}_{i,g}, & \text{otherwise.} \end{cases} \quad (15)$$

The above steps repeat generation by generation until the termination criterion is met.

4 Proposed Algorithm: ATMDE

The performance of COEAs mainly depends on the search ability of evolutionary algorithm and the effectiveness of constraint-handling technique. Hence, the proposed algorithm ATMDE utilizes an improved DE as search optimizer to reproduce offspring and introduces the adaptive trade-off model as the constraint-handling technique to select better individuals to retain into the next population. Furthermore, in order to reduce the redundant search region, the shrinking space technique is employed to enhance the convergence performance. This section will introduce the three core parts of ATMDE algorithm in detail, respectively.

4.1 Improved DE

To balance the convergence rate and accuracy of solution, an improved DE is used as the search engine for the proposed ATMDE algorithm. The set of offspring individuals Q_g is generated by the following three different mutant strategies (i.e. DE/rand/best/1, DE/current-to-rand/1, and DE/rand/2) which combine the advantages of different mutant strategies to generate individuals in order to increase the maximum probability of generating better offspring into evolutionary population. The strategies DE/current-to-rand/1 and DE/rand/2 have the strong explorative ability to generate new promising individuals to add into the evolutionary population, and the strategy DE/rand/best/1 has good exploitative ability to reproduce better individuals around the best individual. A good tradeoff between the global and local search performance can be achieved by combing the three different mutant strategies. The framework of generating offspring is shown in Fig. 2. Each individual in the parent population is employed to generate three different offspring with three different mutant strategies and binomial crossover, and then the better individuals retaining into the next generation are chosen from the new offspring and parent population by ATM strategy.

The implementation of constructing "DE/rand/best/1" strategy is explained as follows. At the beginning, the "DE/rand/1" strategy is introduced to maintain the diversity of population in order to prevent the population from being stuck in a local optimum. This strategy has the ability to enhance the global search ability because the new individuals could learn the information from other individuals randomly chosen from the whole population. Then it is necessary to accelerate the convergence of the evolutionary population, so the "DE/best/1" strategy is employed to speed up convergence as the feasibility proportion of current population increases. The ''DE/best/1'' strategy utilizes the information of the best individual in the current population to generate new individual which can enhance the convergence speed. Hence, the proposed strategy "DE/rand/best/1" as shown in Algorithm 1 is constructed to balance diversity and convergence speed, which combines the "DE/rand/1" strategy and "DE/best/1" strategy through the feasibility proportion of current population. Specially, if a value randomly generated from [0, 1] is greater than the feasibility proportion of current population φ, the "DE/rand/1" strategy is adopted. Otherwise, the "DE/best/1" strategy is employed.

Algorithm 1 The "DE/rand/best/1" strategy if rand(0, 1) $> \varphi$, where φ denotes the feasibility proportion of current population

$\boldsymbol{v}_i = \boldsymbol{x}_{r_1} + F \times (\boldsymbol{x}_{r_2} - \boldsymbol{x}_{r_3})$ #DE/rand/1#

else

$\boldsymbol{v}_i = \boldsymbol{x}_{\text{best}} + F \times (\boldsymbol{x}_{r_1} - \boldsymbol{x}_{r_2})$ #DE/best/1#

end

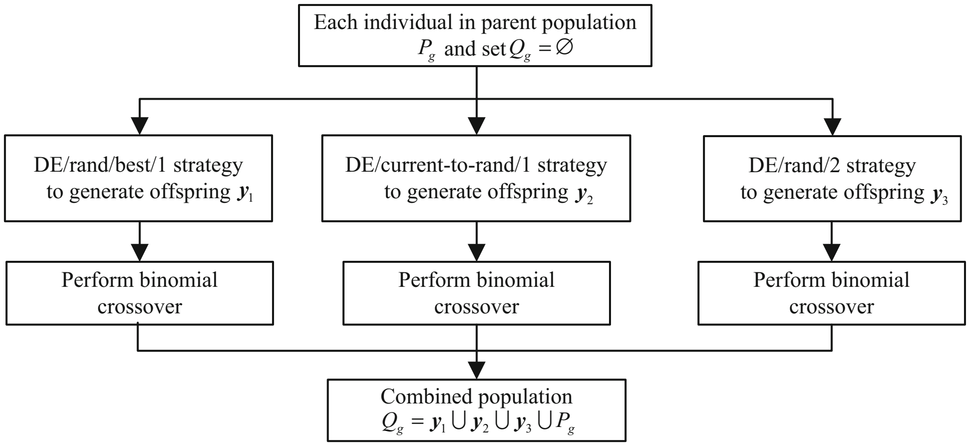

Fig. 2 Flowchart of generating offspring

4.2 Adaptive Trade-Off Model

Generally, a constraint-handling technique to address constraints experiences three different situations in the whole evolutionary process: (1) the infeasible situation only includes the infeasible solutions; (2) the semi-feasible situation includes the feasible and infeasible individuals simultaneously; and (3) the infeasible situation only includes the infeasible individuals. The ATM strategy aims to construct an effective tradeoff scheme to address constraints for each situation according to their corresponding characteristics.

4.2.1 Infeasible Situation

In the infeasible situation, a hierarchical non-dominated selection strategy is introduced to choose individuals from Pareto front into the next population along with evolutionary process and is executed as follows: only the first half of non-dominated individuals with smaller constraint violations are selected to offspring population and are immediately eliminated from the parent population. This process repeats until the number of individuals reaches the size of the offspring population.

4.2.2 Semi-feasible Situation

In this situation, in order to balance the influence of objective value and constraint violation, the adaptive fitness transformation method is employed to calculate the fitness function $f_{\text{fit}}(\boldsymbol{x}_i)$ of individual $\boldsymbol{x}_i$. Firstly, the population is divided into the feasible individual group (Z_1) and infeasible individuals group (Z_2). The objective function value $f(\boldsymbol{x}_i)$ of solution $\boldsymbol{x}_i$ is converted into

$$f'(x_i) = \begin{cases} f(x_i), i \in Z_1, \\ \max\{\varphi f(x_{\text{best}}) + (1-\varphi) f(x_{\text{worst}}), f(x_i)\},\ i \in Z_2, \end{cases} \tag{16}$$

where $f'(\boldsymbol{x}_i)$ is the converted objective function's value of solution $\boldsymbol{x}_i$, and $\boldsymbol{x}_{\text{best}}$ and $\boldsymbol{x}_{\text{worst}}$ are the best and worst solution in the group Z_1, respectively. In order to assign equal importance to different objective functions, it is normalized as

$$f_{\text{nor}}(x_i) = \frac{f'(x_i) - \min_{j \in Z_1 \cup Z_2} f'(x_j)}{\max_{j \in Z_1 \cup Z_2} f'(x_j) - \min_{j \in Z_1 \cup Z_2} f'(x_j)}. \tag{17}$$

The constraint violation value calculated by Eq. (4) is also normalized as

$$G_{\text{nor}}(x_i) = \begin{cases} 0, \quad i \in Z_1, \\ \dfrac{G(x_i) - min_{j \in Z_2} G(x_j)}{max_{j \in Z_2} G(x_j) - min_{j \in Z_2} G(x_j)}, \quad i \in Z_2. \end{cases} \tag{18}$$

Eventually, the final fitness function of solution $\boldsymbol{x}_i$ is calculated by

$$f_{\text{fit}}(x_i) = f_{\text{nor}}(x_i) + G_{\text{nor}}(x_i),\ i \in Z_1 \cup Z_2. \tag{19}$$

The individuals are ranked based on the values of $f_{\text{fit}}(\cdot)$ in ascending order, and the individuals with smaller values are chosen to add into the offspring population until reaching its allowable size.

4.2.3 Feasible Situation

In this feasible situation, the constraint violations of COPs with zero are equivalent to be one of the unconstrained optimization problems because constraint violations of every individual are zero. Hence, only objective function is required to be considered, and Eq. (19) can be also used as a criterion to select better individuals because $G_{\text{nor}}(\cdot)$ is zero.

4.3 Shrinking Space Technique

The shrinking space technique is one of the most pivotal ingredients of IS-PAES [27] and AATM [28]. This technique aims to reduce the search region to focus the computational effort on the specific promising feasible. The main procedure of the shrinking space technique is carried out as Algorithm 2, where T denotes that the technique is performed at every T generations, α_i is a threshold number, β is a reduced factor, and $\bar{x}_{pob,i}$ and $\underline{x}_{pob,i}$ denote the upper and lower bounds of the i-th variable in the selected offspring population, respectively. Afterward, the following specific operations are performed to shrink the search space around the promising individuals to determine the new boundaries for design variables.

Algorithm 2 The shrinking space technique

```
if mod(g, T)=0 then
  for i=1 to n do
    if (x^g_{max,i} − x^g_{min,i}) > α_i then
      width_pob_i = x̄_{pob,i} − x_{pob,i},
      width^g_i = x^g_{max,i} − x^g_{min,i},
      deltaMin_i = (β × width^g_i − width_pob_i)/2,
      delta_i = max(0, deltaMin_i),
      x^{g+1}_{max,i} = x̄_{pob,i} + delta_i and x^{g+1}_{min,i} = x_{pob,i} − delta_i,
      judge whether x^{g+1}_{max,i} and x^{g+1}_{min,i} violate
      boundary constraint,
    end if
  end for
else
    x^{g+1}_max = x^g_max and x^{g+1}_min = x^g_min,
end if
```

4.4 Framework of ATMDE

ATMDE algorithm including an improved differential evolution and adaptive trade-off model and shrinking space technique is constructed to deal with COPs, and the main procedure of ATMDE is shown in Fig. 3. Firstly, it randomly generates NP individuals from the search domain [$\boldsymbol{x}_{\min}$, $\boldsymbol{x}_{\max}$] and then calculates the constraint value $G(\boldsymbol{x})$, the function value $f(\boldsymbol{x})$ and the feasibility proportion of current population φ. Secondly, it generates 3NP offspring individuals from the parent population P_g by the improved DE operator. Thirdly, the better NP individuals are selected from the combined population Q_g into the next generation by ATM strategy, and then the shrinking space technique is employed to reduce the search region to focus the search effort on the promising feasible region when it is satisfied the given condition. Finally, the procedures repeat until meeting the stopping criterion (the maximum generation or the maximum function fitness evaluations, *max_FFEs*).

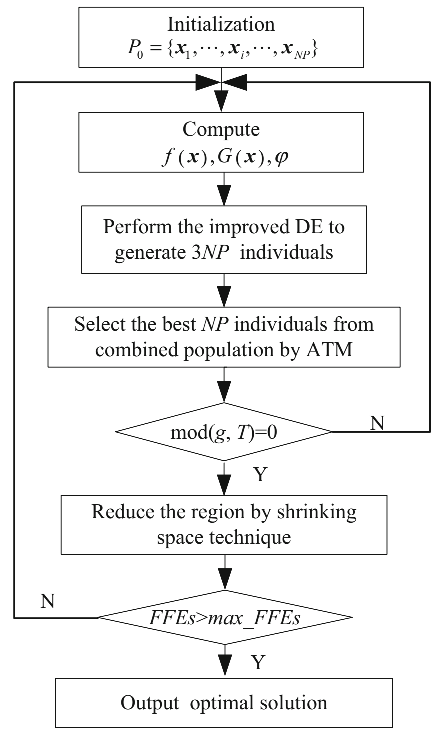

Fig. 3 Framework of ATMDE

5 Benchmark Test Functions

In this part, the performance of the proposed algorithm is verified by 18 benchmark functions from the IEEE CEC2006 [29]. The details of these test functions are listed in Table 1. In this table, $\rho = F/S$ is the estimated ratio between the feasible region and whole search space, where F is the number of feasible ones in S = 1,000,000 randomly generated from search domain [$\boldsymbol{x}_{\min}$, $\boldsymbol{x}_{\max}$]. N denotes the number constraints.

5.1 Parameter Settings

For the numerical simulations, the following parameter settings are utilized unless some changes are mentioned: population size $NP = 50$, tolerance of equality $\delta = 1 \times 10^{-4}$, crossover rate $CR = 0.9$, scaling factor $F = 0.8$, maximum generation $g = 600$. Meanwhile, the parameters of the shrinking space technique are set as:

$$\begin{aligned} &T = 20, \\ &\beta = \sqrt[n]{0.02}, \\ &\alpha_i = \left(x^0_{\max,i} - x_{\min,i}{}^0\right) / \left(20 \times 3^{\lg\left(x_{\max,i}{}^0 - x_{\min,i}{}^0\right)}\right). \end{aligned} \tag{20}$$

To compare the robustness of different algorithms, benchmark functions are optimized at 30 independent runs. Then their statistical performances of the optimal solutions such as mean, standard deviation criteria are utilized to compare.

5.2 General Performance of ATMDE

The numerical results of the eighteen-benchmark functions obtained by the ATMDE algorithm are summarized in Table 2. This table includes the known "optimal" solution for each benchmark function and the "best", "median", "mean", "worst", and "standard deviation" of each test function solved by the ATMDE algorithm. From Table 2, it shows that ATMDE has the strong ability to converge to the global optima for all test functions expect for functions g02 and g19. However, the benchmark functions g02 and g19 solved by ATMDE are extremely close to the known "optimal" values with small standard deviations. The rest sixteen test functions (g01, g03, g04, g05, g06, g07, g08, g09, g10, g11, g12, g14, g15, g16, g18, g24) can be found consistently to achieve "optimal" values in terms of the "best", "median", "mean", "worst", and "standard deviation" criteria. Especially, the functions g01 and g12 can both converge to "optimal" values and their standard deviations are zero, which means that 30 independent runs can obtain their corresponding optimal values.

5.3 ATMDE Compared with AATM

The principal aim of this part verifies that the improved DE as the search optimizer is very effective and can be utilized to further enhance the performance of AATM.

Table 1 Details about 18 benchmark functions

Function	No. of variables n	Type of function	Ration ρ/ %	No. of constraints N
g01	13	Quadratic	0.01	9
g02	20	Nonlinear	99.9	2
g03	10	Polynomial	0.00	1
g04	5	Quadratic	52.1	6
g05	4	Cubic	0.00	5
g06	2	Cubic	0.01	2
g07	10	Quadratic	0.00	8
g08	2	Nonlinear	0.86	2
g09	7	Polynomial	0.51	4
g10	8	Linear	0.00	6
g11	2	Quadratic	0.00	1
g12	3	Quadratic	4.48	1
g14	10	Nonlinear	0.00	3
g15	3	Quadratic	0.00	2
g16	5	Nonlinear	0.02	38
g18	9	Quadratic	0.00	13
g19	15	Nonlinear	33.4	5
g24	2	Linear	79.6	2

To make a fair comparison, the results of benchmark functions by AATM are obtained from the original literature [28]. The comparison results between AATM and ATMDE are listed in Table 3. *w*/*t*/*l* denotes that the proposed ATMDE wins in *w* functions, equals to *t* functions, and loses in *l* functions, compared with AATM algorithm. The results by ATMDE is significantly better than those solved by AATM in 8, 11, 11, and 15 functions in terms of the "best", "mean", "worst", and "standard deviation" criteria, respectively. It equals to its corresponding results in 10, 7, 7, and 1 functions from the above criteria. For the standard deviations, the results only lose in functions g08 and g24, but their differences are extremely close and they both can achieve the "optimal" results with exceedingly small standard deviation. Based on the above comparison, it is clear that ATMDE achieves the competitive better performance than AATM in terms of the quality of the results by these benchmark functions.

Furthermore, the computational cost of ATMDE and AATM both are relatively low compared with the IS-PAES algorithm [27], but the performance of ATMDE is better than those solved by AATM in terms of quality of results. It should be noted that comparison results between AATM and IS-PAES are shown in the reference [28] in which AATM with smaller fitness function evaluations (FFEs) has better performance than IS-PAES. Hence, ATMDE is an effective and efficient algorithm with limited FFEs for solving COPs.

5.4 Effectiveness of the "DE/rand/best/1" Strategy

In order to verify the effectiveness of the proposed "DE/rand/best/1" strategy, 18 test functions are also employed to perform another numerical simulation (i.e. ATMDE1) which only "DE/rand/1" without "DE/best/1" strategy is used to generate the first offspring $\boldsymbol{y}_1$ in the whole search process. For each function, 30 independent runs are also conducted without changing any parameter settings. The comparing results of ATMDE and ATMDE1 summarizes in Table 4. Eleven functions (i.e. g04, g05, g06, g08, g09, g11, g12, g14, g15, g18, g24) can consistently converge to the global optima solved by both ATMDE and ATMDE1. However, the results of the seven functions (i.e. g01, g02, g03, g07, g10, g16, g19) solved by ATMDE can achieve the global optima but the ATMDE1 cannot consistently obtain ones especially for the functions g02, g03 and g10. More specifically, the results achieved by ATMDE are better than those solved by ATMDE1 in 6, 6, 7, 7 and 14 functions in terms of the "best", "median", "mean", "worst", and "standard deviation" criteria, respectively. It ties its corresponding results in 12, 12, 11, 11 and 3 functions from the above criteria. For the standard

Table 2 Results obtained by ATMDE for 18 benchmark test function over 30 independent runs

Function	Optimal solution f^*	Best solution f_{best}	Median solution f_{median}	Mean solution μ_f	Worst solution f_{worst}	Standard deviation σ_f
g01	−15.000	−15.000	−15.000	−15.000	−15.000	0
g02	−0.803 619	−0.803 617	−0.803 617	−0.803 617	−0.803 610	$1.238\,9 \times 10^{-6}$
g03	−1.000 50	−1.005 00	−1.005 00	−1.005 00	−1.005 00	$2.081\,6 \times 10^{-9}$
g04	−30 665.538 6	−30 665.538 6	−30 665.538 6	−30 665.538 6	−30 665.538 6	1.110×10^{-11}
g05	5126.496 71	5126.496 71	5126.496 71	5126.496 71	5126.496 71	$1.013\,3 \times 10^{-12}$
g06	−6961.813 87	−6961.813 87	−6961.813 87	−6961.813 87	−6961.813 87	1.850×10^{-12}
g07	24.306 209	24.306 209	24.306 209	24.306 209	24.306 209	$2.211\,4 \times 10^{-8}$
g08	−0.095 825	−0.095 825	−0.095 825	−0.095 825	−0.095 825	$2.564\,1 \times 10^{-17}$
g09	680.630 05	680.630 05	680.630 05	680.630 05	680.630 05	$4.634\,8 \times 10^{-13}$
g10	7049.248 02	7049.248 02	7049.248 02	7049.248 02	7049.24802	8.700×10^{-7}
g11	0.749 90	0.749 90	0.749 90	0.749 90	0.749 90	$1.011\,2 \times 10^{-7}$
g12	−1.000 00	−1.000 00	−1.000 00	−1.000 00	−1.000 00	0
g14	−47.764 888	−47.764 888	−47.764 888	−47.764 888	−47.764 888	$1.953\,9 \times 10^{-10}$
g15	961.715 022	961.715 022	961.715 022	961.715 022	961.715 022	$6.937\,8 \times 10^{-13}$
g16	−1.905 155	−1.905 155	−1.905 155	−1.905 155	−1.905 155	$6.775\,2 \times 10^{-16}$
g18	−0.866 025	−0.866 025	−0.866 025	−0.866 025	−0.866 025	$7.454\,9 \times 10^{-10}$
g19	32.655 59	32.655 63	32.655 86	32.656 00	32.657 25	$3.753\,8 \times 10^{-4}$
g24	−5.508 013	−5.508 013	−5.508 013	−5.508 013	−5.508 013	$3.735\,5 \times 10^{-15}$

Table 3 Comparison results of ATMDE and AATM on 18 benchmark test functions

Function	Best solution f_{best}		Mean solution μ_f		Worst solution f_{worst}		Standard deviation σ_f	
	ATMDE	AATM	ATMDE	AATM	ATMDE	AATM	ATMDE	AATM
g01	−15.000	−15.000	−15.000	−15.000	−15.000	−15.000	0	3.1×10^{-7}
g02	−0.803 617	−0.803 38	−0.803 617	−0.791 21	−0.803 61	−0.767	1.2×10^{-6}	8.6×10^{-3}
g03	−1.005 00	−1.00	−1.005 00	−1.00	−1.005 00	−1.00	2.1×10^{-9}	3.5×10^{-4}
g04	−30 665.539	−30 665.5	−30 665.539	−30 665.5	−30 665.5	−30 665.5	1.1×10^{-11}	1.0×10^{-4}
g05	5 126.496 7	5 126.498	5 126.496 71	5 126.714	5 126.496 7	5 128.824	1.0×10^{-12}	4.3×10^{-1}
g06	−6 961.814	−6 961.81	−6 961.814	−6 961.81	−6 961.81	−6 961.81	1.6×10^{-12}	7.1×10^{-12}
g07	24.306 209	24.307	24.306 209	24.317	24.306 209	24.356	2.2×10^{-8}	1.3×10^{-2}
g08	−0.095 825	−0.095 82	−0.095 825	−0.095 82	−0.095 82	−0.095 82	2.6×10^{-17}	5.8×10^{-18}
g09	680.630	680.630	680.630 05	680.639 4	680.630 05	680.646	4.6×10^{-13}	4.5×10^{-3}
g10	7 049.248	7 049.603	7 049.2480 2	7 077.477	7 049.248	7 183.295	8.7×10^{-7}	3.1×10^{1}
g11	0.74990	0.75	0.7499	0.75	0.7499	0.75	1.0×10^{-7}	3.8×10^{-6}
g12	−1.000	−1.000	−1.000	−1.000	−1.000	−1.000	0	0
g14	−47.764 888	−47.762	−47.764 888	−47.750	−47.764 8	−47.712	1.9×10^{-10}	1.0×10^{-2}
g15	961.715	961.715	961.715	961.715	961.715 02	961.716	6.9×10^{-13}	3.0×10^{-4}
g16	−1.905 155	−1.905 15	−1.905 155	−1.905 15	−1.905 15	−1.905 15	6.8×10^{-16}	2.4×10^{-14}
g18	−0.866 025	−0.866 02	−0.866 025	−0.865 95	−0.866 02	−0.864 84	7.5×10^{-10}	2.1×10^{-4}
g19	32.655 63	32.725	32.655 86	32.952	32.657 25	33.243	3.8×10^{-4}	1.4×10^{-1}
g24	−5.508 01	−5.508 01	−5.508 01	−5.508 01	−5.508 01	−5.508 01	3.7×10^{-15}	1.8×10^{-15}
w/*t*/*l*	8/10/0		11/7/0		11/7/0		15/1/2	

deviation criterion, the function g11 solved by ATMDE1 is smaller than one by ATMDE, but they both can obtain the optimal result with an exceedingly small difference. Based on the above analysis, the proposed "DE/rand/best/1" strategy is a very important part of the proposed ATMDE algorithm.

Table 4 Results obtained by ATMDE and ATMDE1 on 18 benchmark test functions

Function	Method	Best solution f_{best}	Median solution f_{median}	Mean solution μ_f	Worst solution f_{worst}	Standard deviation σ_f
g01	ATMDE	−15.000	−15.000	−15.000	−15.0000	0
	ATMDE1	−14.999 9	−14.999 9	−14.999 9	−14.999 9	8.98×10^{-7}
g02	ATMDE	−0.803 617	−0.803 617	−0.803 617	−0.803 610	1.24×10^{-6}
	ATMDE1	−0.802 125	−0.802 125	−0.802 124	−0.802 092	6.14×10^{-6}
g03	ATMDE	−1.005 00	−1.005 00	−1.005 00	−1.005 00	2.08×10^{-9}
	ATMDE1	−1.005 00	−1.005 00	−0.985 8	−0.798 4	4.93×10^{-2}
g04	ATMDE	−30 665.53	−30 665.53	−30 665.53	−30 665.5	1.11×10^{-11}
	ATMDE1	−30 665.53	−30 665.53	−30 665.53	−30 665.53	1.85×10^{-11}
g05	ATMDE	5126.496 71	5 126.496 71	5126.496 71	5 126.496 71	1.01×10^{-12}
	ATMDE1	5126.496 71	5 126.496 71	5126.496 71	5 126.496 71	2.95×10^{-9}
g06	ATMDE	−6961.813	−6961.813	−6961.813	−6961.81	1.85×10^{-12}
	ATMDE1	−6961.813	−6961.813	−6961.813	−6961.81	2.78×10^{-12}
g07	ATMDE	24.306 209	24.306 209	24.306 209	24.306 209	2.21×10^{-8}
	ATMDE1	24.3062 497	24.306 2497	24.306 253	24.306 364	2.09×10^{-5}
g08	ATMDE	−0.095 825	−0.095 825	−0.095 825	−0.095 825	2.56×10^{-17}
	ATMDE1	−0.095 825	−0.095 825	−0.095 825	−0.095 825	2.82×10^{-17}
g09	ATMDE	680.630 05	680.630 05	680.630 05	680.630 05	4.63×10^{-13}
	ATMDE1	680.630 05	680.630 05	680.630 05	680.630 05	4.85×10^{-13}
g10	ATMDE	7 049.248 02	7 049.248 02	7 049.248 02	7 049.248 02	8.70×10^{-7}
	ATMDE1	7 049.339 29	7 049.699 7	7 049.800 6	7 051.246 3	4.59×10^{-1}
g11	ATMDE	0.749 90	0.749 90	0.749 90	0.749 90	1.01×10^{-7}
	ATMDE1	0.749 90	0.749 90	0.749 90	0.749 90	1.12×10^{-16}
g12	ATMDE	−1.000 00	−1.000 00	−1.000 00	−1.000 00	0
	ATMDE1	−1.000 00	−1.000 00	−1.000 00	−1.000 00	0
g14	ATMDE	−47.764 888	−47.764 888	−47.764 888	−47.764 88	1.95×10^{-10}
	ATMDE1	−47.764 888	−47.764 888	−47.764 888	−47.764 88	1.67×10^{-8}
g15	ATMDE	961.715 022	961.715 022	961.715 022	961.715 022	6.94×10^{-13}
	ATMDE1	961.715 022	961.715 022	961.715 022	961.715 022	6.94E $\times 10^{-13}$
g16	ATMDE	−1.905 155	−1.905 155	−1.905 155	−1.905 155	6.78×10^{-16}
	ATMDE1	−1.905 102	−1.905 102	−1.905 102	−1.905 102	6.78×10^{-16}
g18	ATMDE	−0.866 025	−0.866 025	−0.866 025	−0.866 025	7.45×10^{-10}
	ATMDE1	−0.866 025	−0.866 025	−0.866 025	−0.866 025	4.49×10^{-6}
g19	ATMDE	32.655 63	32.655 86	32.656 00	32.657 25	3.75×10^{-4}
	ATMDE1	32.676 38	32.702 88	32.704 75	32.774 96	2.16×10^{-2}
g24	ATMDE	−5.508 013	−5.508 013	−5.508 013	−5.508 013	3.74×10^{-15}
	ATMDE1	−5.508 013	−5.508 013	−5.508 013	−5.508 013	4.52×10^{-15}
	w/t/l	6/12/0	6/12/0	7/11/0	7/11/0	14/3/1

5.5 Four Mechanical Benchmark Engineering Designs

The four mechanical benchmark engineering problems are used by many researchers to demonstrate the performance of different algorithms. Different characteristics of objective functions and constraints are illustrated as shown in Table 5. The four mechanical engineering designs [28] are the minimum cost of a weld-beam design, the minimum weight of a spring design, the minimum weight of a speed reducer design and the minimum volume of a three-bar truss design, respectively. Table 6 summarizes the comparative results of these problems solved by ATMDE and AATM in terms of the "best", "mean", "worst", and "standard deviation" metrics. It shows that the ATMDE has the better statistically quality and robustness than AATM under the same number of function fitness evaluations in terms of the selected performance criteria.

Moreover, the four standard deviations obtained by ATMDE are relatively small, which is a crucial feature for application of the approach for solving the practical world problems. Table 7 lists the best design variables obtained by ATMDE and AATM for four engineering design problems associated with their corresponding optimal results, which show the ATMDE algorithm can obtain better solution than the AATM.

6 Engineering Applications

6.1 Vehicle Crashworthiness Problem

In the automotive industry, structural optimization design for vehicle crashworthiness has become a paramount research field. In this paper, different characteristics of low- and high-speed crashworthiness are considered simultaneously [30]. For the frontal impact, the most crucial energy absorbing components including rail, collision beam and stiffener can directly affect the performance of vehicle crashworthiness and safety. Therefore, the total mass $M(\boldsymbol{x})$ of selected parts including collision beam, stiffener, front rail and front rail cover as shown in Fig. 4 is considered as our optimization objective and it is also subjected to acceleration, energy-absorbing and maximum intrusion constraints. Then, the crashworthiness problem could be formulated as

$$\min f(x) = M(x),\quad \text{s.t.}\begin{cases} g_1(x) = \bar{a}(x) - 35 \le 0,\\ g_2(x) = E(x) - 300 \le 0,\\ g_3(x) = I_{\text{up}}(x) - 350 \le 0,\\ g_4(x) = I_{\text{down}}(x) - 200 \le 0, \end{cases} \tag{21}$$

where $\boldsymbol{x} = [x_1, x_2, x_3, x_4, x_5]$, $2\text{ mm} \le x_1 \le 3\text{ mm}$, $1\text{ mm} \le x_2, x_3 \le 2.5\text{ mm}$, $1.5\text{ mm} \le x_4, x_5 \le 3\text{ mm}$, $\bar{a}(\boldsymbol{x})$ is the mean value of integral acceleration, $E(\boldsymbol{x})$ is the energy-absorbing of inner and outer front rail, $I_{\text{up}}(\boldsymbol{x})$ and $I_{\text{down}}(\boldsymbol{x})$ are the intrusion of the two points at the engine as shown in Fig. 5, respectively.

The finite element model (FEM) of the vehicle including 755 parts and 977 742 elements is established for the above objective and constraints. To improve efficiency, the response surfaces are established based on the samples by Latin hypercube sampling method. Then, the minimum mass $M(\boldsymbol{x})$ solved by ATMDE algorithm is 10.53 kg and its corresponding five design variables are 2.00 mm, 2.50 mm, 2.50 mm, 2.76 mm and 1.68 mm, respectively. Under this circumstance, values of constraints are 0 g, −4208.33 J, − 60.74 mm, −0.0002 mm, respectively. Specifically, the mean value of integral acceleration $\bar{a}(\boldsymbol{x})$ is 35 g, which can effectively protect passengers in the automobile when the collision inevitably occurs. Meanwhile, the inner and outer front rail can absorb 3908.33 J. In addition, the intrusions of upper and lower point at the engine are 289.26 mm and 200 mm, which can effectively reduce the occupants' injuries to protect the passengers' safety.

Table 5 Main features for each engineering design problem

Engineering benchmark	No. of variables n	Ration ρ/ %	No. of constrains N
Weld-beam design	4	37.625	5
Spring design	3	0.732 3	4
Speed reducer design	7	23.015 2	11
Three-bar truss design	2	21.870 6	3

6.2 Structural Optimization Design of Tablet Computer

Currently, Tablet computer is one of typical popular consumer electronic devices, which have high-integrated density and large power dissipation. It is inevitably for its structural design to consider various aspects of design requirements, such as appearance, portability, operating

Table 6 Results about four benchmark engineering design problems

Engineering problems	Method	Best solution f_{best}	Mean solution μ_f	Worst solution f_{worst}	Standard deviation σ_f
Weld-beam design	ATMDE	2.380 956	2.380 956	2.380 956	5.88×10^{-11}
	AATM	2.382 326	2.386 976	2.391 592	2.20×10^{-3}
Spring design	ATMDE	0.012 665	0.012 665	0.012 665	1.05×10^{-15}
	AATM	0.012 668	0.012 708	0.012 861 37	4.50×10^{-5}
Speed reducer	ATMDE	2994.473 6	2994.474 4	2994.474 45	1.18×10^{-5}
	AATM	2994.516 7	2994.585 4	2994.659 79	3.30×10^{-2}
Three-bar truss design	ATMDE	263.895 84	263.895 84	263.895 843	2.87×10^{-13}
	AATM	263.895 84	263.896 6	263.900 41	1.10×10^{-3}

Table 7 Best design variables for four benchmark engineering design problems

Engineering problems	Method	Best design variable x_{best}	Best function values f_{best}
Weld-beam design	ATMDE	0.244 368 975, 6.217 519 715, 8.291 471 390, 0.244 368 975	2.380 956 580
	AATM	0.244 106 586, 6.220 903 633, 8.298 161 229,0.244 382 231	2.382 326
Spring design	ATMDE	0.356 717 739, 0.051 689 061, 11.288 965 783 04	0.012 665 232
	AATM	0.359 690 411, 0.051 813 095, 11.119 252 680	0.012 668 261
Speed reducer design	ATMDE	3.50, 0.7, 17, 7.309 819 903, 7.715 173 384 44, 3.350 233 018 67, 5.286 521 228 48	2 994.473 624
	AATM	3.500 016 221, 0.700 001 177, 17.000 029 883, 7.300 297 290, 7.716 049 465, 3.350 239 798, 5.286 660 476 6	2 994.516 778
Three-bar truss design	ATMDE	0.788 675 135, 0.408 248 289	263.895 843
	AATM	0.788 681 755, 0.408 229 565	263.895 843

safety, etc. Hence, structural optimization design of tablet computer should be guaranteed to work well under different conditions. This subsection considers the structural optimization design of a 7-inches tablet computer, as illustrated in Fig. 6 which mainly includes the touch screen, the display, the battery, the mainboard, the inner bracket, the front shell and the back shell [31]. Our optimization objective mainly considers the minimization of tablet's thickness $D(\boldsymbol{x})$. The design problem should be satisfied four practical work conditions including high-temperature constraint $g_1(\boldsymbol{x})$, room-temperature constraint $g_2(\boldsymbol{x})$, and alternating temperature constraint $g_3(\boldsymbol{x})$ and free fall constraint $g_4(\boldsymbol{x})$. Hence, this optimization problem could be formulated as

$$\min f(x) = D(x),\\ \text{s.t.} \begin{cases} g_1(x) = T^{CH}(x) - 65 \le 0, \\ g_2(x) = T^{SH}(x) - 40 \le 0, \\ g_3(x) = \Gamma^{BA}(x) - 24 \le 0, \\ g_4(x) = \Gamma^{TS}(x) - 100 \le 0, \end{cases} \tag{22}$$

where $T^{CH}(\boldsymbol{x})$ denotes the temperature of the chip on the main board under the high-temperature (45 °C); $T^{SH}(\boldsymbol{x})$ is the shell surface temperature with a full load under the room temperature (25 °C) for an hour continuously work; $\Gamma^{BA}(\boldsymbol{x})$ is the thermal stress of the battery and $\Gamma^{TS}(\boldsymbol{x})$ is the maximal stress of the touch screen under the collision of the 0.5 m-height free fall. The design variables are the thickness of the front shell x_1, the thickness of the display x_2, the thickness of the inner bracket x_3, the thickness of the back shell x_4, respectively. And their design values should be restricted to the domains $4\text{ mm} \le x_1 \le 6\text{ mm}$, $0.5\text{ mm} \le x_2, x_3, x_4 \le 2\text{ mm}$.

Four finite element models (FEM) are constructed for the above four performance constraints. To improve efficiency, the four corresponding response surfaces are established based on the given samples. Furthermore, the accuracy of the response surfaces is verified. Then the

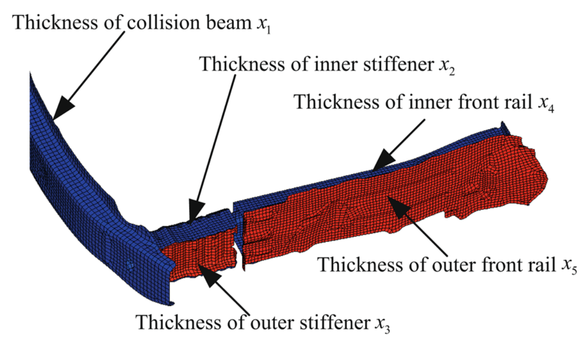

Fig. 4 Selected design variables

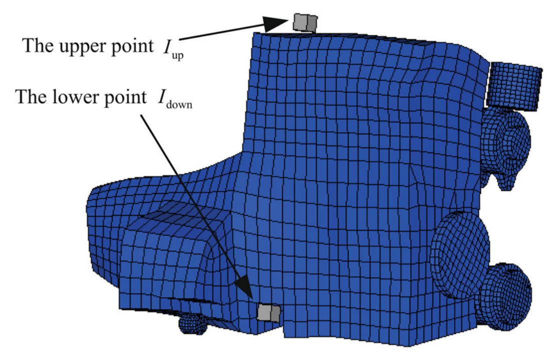

Fig. 5 Intrusion measured by the selected points of engine

ATMDE algorithm is utilized to solve the tablet computer optimization problem. The structural thickness of optimized tablet computer is 6.42 mm which is a 31.7% reduction in compared with that of the original design (6.00 mm, 1.20 mm, 1.20 mm, 1.00 mm), and its design variables are 4.00 mm, 0.51 mm, 1.41 mm and 0.50 mm, respectively. Under this circumstance, the temperature of the chip is 62.05 °C and the temperature of shell surface is 37.66 °C which can ensure consumer daily-using comfortably. The thermal stress of battery is about

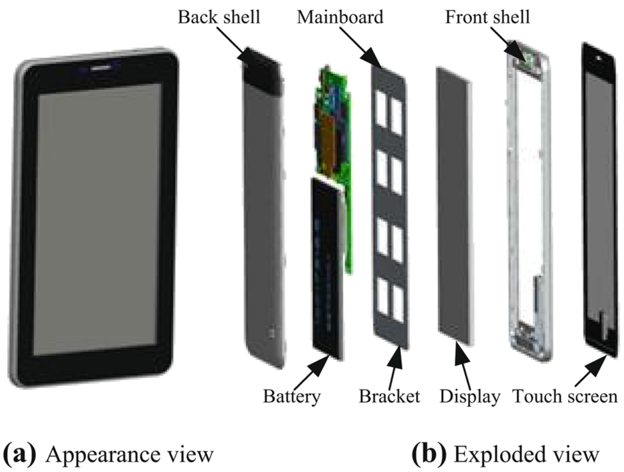

Fig. 6 7-inch tablet computer [31]

24 MPa, which can make sure the operating safety in daily. The maximal stress of touch screen is about 100 MPa, which can avoid the device broken during the collision of 0.5 m free fall. This optimized structural design is meaningful because the consumers are satisfied with the final design with better the appearance and portability for the tablet.

7 Conclusions

(1) An improved differential evolution with shrinking space technique and adaptive trade-off model, named ATMDE, is proposed to solve constrained optimization problems with high accuracy and robustness.

(2) The new "DE/rand/best/1" mutant strategy is constructed to generate offspring by the feasibility proportion of the current population, which could enhance performance of the ATMDE illustrated by results of test functions.

(3) In comparison with AATM algorithm, ATMDE achieves better performance verified by the simulation results of eighteen benchmark test functions from the IEEE CEC2006.

(4) The ATMDE is employed to optimize the structural optimization design of tablet computer, and the optimized thickness is a 31.7% reduction in compared with that of the original design.

References

1. HULL P V, TINKER M L, DOZIER G. Evolutionary optimization of a geometrically refined truss[J]. *Structural and Multidisciplinary Optimization*, 2006, 31(4): 311–319.
2. ZHAO Y, CHEN G, WANG H, et al. Optimum selection of mechanism type for heavy manipulators based on particle swarm optimization method[J]. *Chinese Journal of Mechanical Engineering*, 2013, 26(4): 763–770.
3. TANG Q, LI Z, ZHANG L, et al. Effective hybrid teaching-learning-based optimization algorithm for balancing two-sided assembly lines with multiple constraints[J]. *Chinese Journal of Mechanical Engineering*, 2015, 28(5): 1067–1079.
4. WANG Y, LIU H, CAI Z, et al. An orthogonal design based constrained evolutionary optimization algorithm[J]. *Engineering Optimization*, 2007, 39(6): 715–736.
5. WANG L, LI L. An effective differential evolution with level comparison for constrained engineering design[J]. *Structural and Multidisciplinary Optimization*, 2010, 41(6): 947–963.
6. CHEN D N, ZHANG R X, YAO C Y, et al. Dynamic topology multi force particle swarm optimization algorithm[J]. *Chinese Journal of Mechanical Engineering*, 2016, 29(1): 124–135.
7. RAO R, SAVSANI V, BALIC J. Teaching-learning-based optimization algorithm for unconstrained and constrained real-parameter optimization problems[J]. *Engineering Optimization*, 2012, 44(12): 1447–1462.
8. QUARANTA G, FIORE A, MARANO G C. Optimum design of prestressed concrete beams using constrained differential evolution algorithm[J]. *Structural and Multidisciplinary Optimization*, 2014, 49(3): 441–453.
9. ZHAO Y, WANG H, WANG W, et al. New hybrid parallel algorithm for variable-sized batch splitting scheduling with alternative machines in job shops[J]. *Chinese Journal of Mechanical Engineering*, 2010, 23(4): 484–495.
10. COELLO C A C. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art[J]. *Computer Methods in Applied Mechanics and Engineering*, 2002, 191(11): 1245–1287.
11. TESSEMA B, YEN G G. An adaptive penalty formulation for constrained evolutionary optimization[J]. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans,* 2009, 39(3): 565–578.
12. DEB K, DATTA R. A bi-objective constrained optimization algorithm using a hybrid evolutionary and penalty function approach[J]. *Engineering Optimization*, 2013, 45(5): 503–527.
13. DEB K. An efficient constraint handling method for genetic algorithms[J]. *Computer Methods in Applied Mechanics and Engineering*, 2000, 186(2): 311–338.
14. BALAMURUGAN R, RAMAKRISHNAN C, SWAMINATHAN N. A two phase approach based on skeleton convergence and geometric variables for topology optimization using genetic algorithm[J]. *Structural and Multidisciplinary Optimization*, 2011, 43(3): 381–404.
15. Venkatraman S, YEN G G. A generic framework for constrained optimization using genetic algorithms[J]. *IEEE Transactions on Evolutionary Computation,* 2005, 9(4): 424–435.
16. JIAO L, LI L, SHANG R, et al. A novel selection evolutionary strategy for constrained optimization[J]. *Information Sciences*, 2013, 239: 122–141.
17. STORN R, PRICE K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces[J]. *Journal of Global Optimization*, 1997, 11(4): 341–359.
18. STORN R. System design by constraint adaptation and differential evolution[J]. *IEEE Transactions on Evolutionary Computation,* 1999, 3(1): 22–34.

19. LANDA B R, COELLO C A C. Cultured differential evolution for constrained optimization[J]. *Computer Methods in Applied Mechanics and Engineering*, 2006, 195(33): 4303–4322.
20. MEZURA-MONTES E, COELLO C A C, VEL Z J, et al. Multiple trial vectors in differential evolution for engineering design[J]. *Engineering Optimization*, 2007, 39(5): 567–589.
21. ZHANG M, LUO W, WANG X. Differential evolution with dynamic stochastic selection for constrained optimization[J]. *Information Sciences*, 2008, 178(15): 3043–3074.
22. MALLIPEDDI R, SUGANTHAN P N. Ensemble of constraint handling techniques[J]. *IEEE Transactions on Evolutionary Computation,* 2010,14(4): 561–579.
23. ELSAYED S M, SARKER R A, ESSAM D L. Multi-operator based evolutionary algorithms for solving constrained optimization problems[J]. *Computers & Operations Research*, 2011, 38(12): 1877–1896.
24. GONG W, CAI Z, LIANG D. Engineering optimization by means of an improved constrained differential evolution[J]. *Computer Methods in Applied Mechanics and Engineering*, 2014, 268: 884–904.
25. WANG C, FANG Y, GUO S. Multi-objective optimization of a parallel ankle rehabilitation robot using modified differential evolution algorithm[J]. *Chinese Journal of Mechanical Engineering*, 2015, 28(4): 702–715.
26. WANG Y, CAI Z, ZHOU Y, et al. An adaptive tradeoff model for constrained evolutionary optimization[J]. *IEEE Transactions on Evolutionary Computation,* 2008, 12(1): 80–92.
27. AGUIRRE A H, RIONDA S B, COELLO C A C, et al. Handling constraints using multiobjective optimization concepts[J]. *International Journal for Numerical Methods in Engineering*, 2004, 59(15): 1989–2017.
28. WANG Y, CAI Z, ZHOU Y. Accelerating adaptive trade-off model using shrinking space technique for constrained evolutionary optimization[J]. *International Journal for Numerical Methods in Engineering*, 2009, 77(11): 1501–1534.
29. LIANG J J, RUNARSSON T, MEZURA-MONTES E, et al. Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization[R]. Technical report, Nanyang Technological University, Singapore. 2006.
30. JIANG C, DENG S L. Multi-objective optimization and design considering automotive high-speed and low-speed crashworthiness[J]. *Chinese Journal of Computational Mechanics*, 2014, 31(4): 474–479. (in Chinese)
31. HUANG Z, JIANG C, ZHOU Y, et al. Reliability-based design optimization for problems with interval distribution parameters[J]. Structural and Multidisciplinary Optimization, 2016, 1–16, doi:10.1007/s00158-016-1505-3.

Chunmming FU, born in 1987, is currently a PhD candidate at *State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, Hunan University, China.* His research interests include structural optimization design, differential evolution and constrained optimization. E-mail: fuchunming@hnu.edu.cn

Yadong XU, born in 1978, is currently an associate professor at *Nanjing University of Science and Technology*, China. His research interests include structural optimization design and composite material. E-mail: 13601580051@139.com

Chao JIANG, born in 1978, is currently a professor and PhD candidate supervisor at *Hunan University, China*. His research interests include mechanical design, uncertainty analysis and reliability. Tel: +86-731-88821748; E-mail: jiangc@hnu.edu.cn

Xu HAN, born in 1968, is currently a professor and a PhD candidate supervisor at *Hunan University, China*. His research interests include mechanical design, uncertainty analysis and reliability. E-mail: hanxu@hnu.edu.cn

Zhiliang HUANG, born in 1980, is a PhD candidate at *State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, Hunan University, China*. His research interests include reliability based design optimization. E-mail: 13787181710@163.com