

Research Article

Study on Parameter Optimization Design of Drum Brake Based on Hybrid Cellular Multiobjective Genetic Algorithm

Yi Zhang, Hu Zhang, and Chao Lu

Hubei Key Laboratory of Hydroelectric Machinery Design & Maintenance, China Three Gorges University, Yichang 44300, China

Correspondence should be addressed to Yi Zhang, jxzhangyi1976@126.com

Received 19 July 2012; Revised 14 October 2012; Accepted 15 October 2012

Academic Editor: Jyh Horng Chou

Copyright © 2012 Yi Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In consideration of the significant role the brake plays in ensuring the fast and safe running of vehicles, and since the present parameter optimization design models of brake are far from the practical application, this paper proposes a multiobjective optimization model of drum brake, aiming at maximizing the braking efficiency and minimizing the volume and temperature rise of drum brake. As the commonly used optimization algorithms are of some deficiency, we present a differential evolution cellular multiobjective genetic algorithm (DECell) by introducing differential evolution strategy into the canonical cellular genetic algorithm for tackling this problem. For DECell, the gained Pareto front could be as close as possible to the exact Pareto front, and also the diversity of nondominated individuals could be better maintained. The experiments on the test functions reveal that DECell is of good performance in solving high-dimension nonlinear multiobjective problems. And the results of optimizing the new brake model indicate that DECell obviously outperforms the compared popular algorithm NSGA-II concerning the number of obtained brake design parameter sets, the speed, and stability for finding them.

1. Introduction

The safe reliability of the vehicles is attracting more and more attentions with the sharp increase in the vehicles. Since the improvement of the brake design and manufacturing is of great significance in ensuring the brake quality thus the vehicle's safe running, the brake parameter optimization design which is a key part in the whole brake design procedure is becoming increasingly important. Therefore, the research in this field is becoming a hot topic.

In general, in the brake parameters optimizing procedure, the first step is to set up parameter optimization model and the next step is to calculate the model with an optimization algorithm. A lot of researches have been done in this field [1–13], most of which are

done by Chinese scholars. Their researches mainly focus on two aspects: the establishment of the optimization model [1, 3, 6, 7] and the design, selection, and application of the algorithms for solving the model [2, 4, 9, 10]. Although plenty of achievements have been gained in the existing studies, they still need to be improved to some extent, especially in the following three aspects. First, the complexity of the model is not satisfactory. In general, there are a lot of brake performance indicators, such as brake temperature rise, brake drum volume, and braking efficiency. However, in existing models, only one or two of them are selected as optimization objectives, which is far from engineering practice. Second, the multiobjective optimization problems are often solved by being transferred into single-objective problems with weighting or separate calculating method, which ignores the fact that the solution of a multiobjective optimization problem is a set. Third, the performance of the adopted algorithms cannot meet the requirements. The frequently used algorithms in the existing studies are the MATLAB optimization toolbox, basic genetic algorithms, basic ant colony algorithm, and so forth. However, these algorithms are easy to be trapped in local convergence when solving complex problems, for which the gained results are undesirable.

Based on foregoing statement, this paper establishes a multiobjective optimization design model of drum brake, which aims at maximizing the braking efficiency and minimizing the volume and temperature rise of the drum brake. It is expected that this model could meet the requirements of practical engineering design better. Besides, in order to deal with the model more efficiently, a differential evolution cellular multiobjective genetic algorithm (DECell) is proposed by introducing a differential evolution strategy into the cellular genetic algorithm in which an individual may only interact with individuals belonging to its neighborhood. The proposed algorithm is then applied to solve the brake model.

The rest of the paper is organized as follows. The next section shows the details of the multiobjective optimization model of drum brake. Section 3 presents the differential cellular multiobjective genetic algorithm (DECell). Section 4 tests the performance of DECell using several representative test functions. The optimization procedures of design parameters of new brake model are presented in Section 5. And in Section 6, we study a case of designing brake using DECell. This paper concludes with a summary in Section 7.

2. Multiobjective Optimization Design Model of Drum Brake

Drum brake is one of the most commonly used brakes in vehicle design; it can be categorized into leading- and trailing-shoe brake, two-leading-shoe brake, two-trailing-shoe brake, and servo brake concerning the arrangement of the brake shoes. The optimization design object in this paper is the leading- and trailing-shoe brake, which is shown in Figure 1.

Figure 1 shows the structural parameters and force diagram of the leading- and trailing-drum brake (only left part of the structure is presented). With the effect of braking force P , the two brake shoes at both sides rub the drum brake to generate frictional resisting torque and thus to brake. In Figure 1, F_1 is the pressing force of the brake shoe, while R_f is the action radius of the frictional force. δ represents the angle between x axis and the active line of F_1 . The main structural parameters are the starting angle of brake friction plate (θ_0), wrap angle of friction plate (θ), the distance from braking force P to the brake center (a), the central location of the brake shoe pin (c, c'), and so forth.

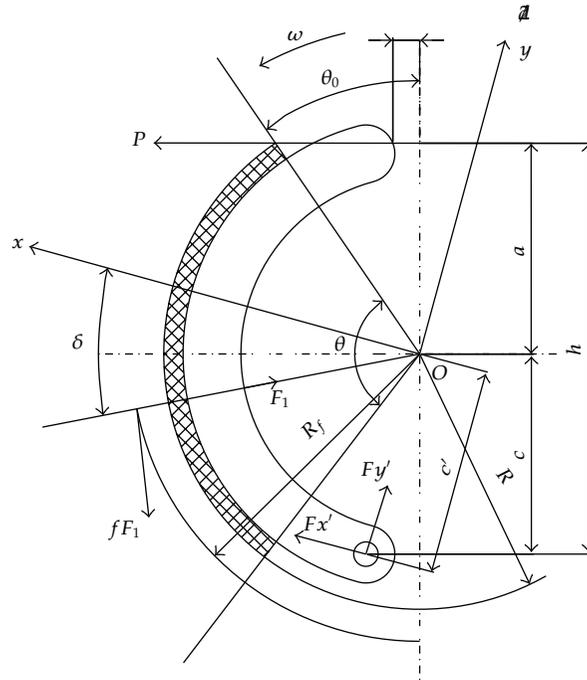


Figure 1: Structural parameters and force diagram.

2.1. Design Variables

One has $X^T = [x_1, x_2, \dots, x_{11}] = [\theta_0, \theta, b_1, R, c', c, a, b_2, e, d, p]$, where θ_0 is the starting wrap angle of the friction plate, θ is the wrap angle of the friction plate, b_1 is the friction plate width, R is the brake drum radius, c and c' are the central sites of the brake shoe pin, a means the distance from the braking force to the brake center, b_2 is the brake width, e is the thickness of the brake drum, d is the brake cylinder diameter, and p is the braking oil pressure.

2.2. Objective Functions

The optimization target is the comprehensive performance of the brake. And the subobjectives are maximizing the braking efficiency factor and minimizing the volume and temperature rise of the drum brake. The subobjectives are presented as follows.

(1) Braking Efficiency Factor

One has the following [14]:

$$K = \frac{M_\mu}{PR_f}, \tag{2.1}$$

where M_μ is the braking output torque, P is the input force, and R_f represents the action radius of the frictional force. Therefore, efficiency factor of the brake with tight (loose) shoe is as follows:

$$K_{1,2} = \frac{\mu(a+c)}{c'(\cos \delta \pm \mu \sin \delta) \mp \mu R_f}, \quad (2.2)$$

where

$$R_f = \frac{4R[\cos \theta_0 - \cos(\theta_0 + \theta)]}{\sqrt{[\cos 2\theta - \cos 2(\theta_0 + \theta)]^2 + [2\theta - \sin 2(\theta_0 + \theta) + \sin 2\theta_0]^2}}, \quad (2.3)$$

$$\delta = \arctan \left[\frac{\cos 2\theta_0 - \cos 2(\theta_0 + \theta)}{2\theta - \sin 2(\theta_0 + \theta) + \sin 2\theta_0} \right].$$

μ is the friction factor between the brake drum and the friction plate.

With the same braking force, the bigger the braking efficiency is, the bigger the braking torque is, and the better the brake performs, hence the more efficient the brake is. Therefore, the enhancement of braking efficiency is of great significance to the safe running of vehicle. On this basis, the braking efficiency factor is selected as the optimization objective, and the objective function is

$$f_1(X) = K_1 + K_2, \quad (2.4)$$

where K_1 represents efficiency factor of the brake with tight shoe and K_2 represents efficiency factor of the brake with loose shoe (see (2.2)).

(2) The Volume of the Drum Brake

With the same function and efficiency, the smaller the brake is, the fewer materials and the less space it will take up. Hence, the volume of the brake drum should be as small as possible, and it is taken as the optimization objective as follows:

$$f_2(X) = V = \pi b_2(R+e)^2. \quad (2.5)$$

(3) Temperature Rise of Drum Brake

It can be regarded that the heat energy generated through braking is absorbed by the front brake and rear brake due to the fast braking process. The heat energy is then distributed to the front brake and rear brake in accordance with their distribution coefficients. The brake drum is installed at the rear wheel, so the rear brake temperature can be calculated by the following formula:

$$f_3(X) = \Delta t = \frac{1}{2} \times \frac{m_a v_a^2}{(m_d c_d + m_h c_h) \times 10^{-9}} (1 - \beta), \quad (2.6)$$

where m_a is the total mass of the vehicle (kg); m_d is the total mass of each drum brake (kg); m_h is the total mass of the heating metal materials linking with the drum brake; C_d is the specific heat capacity (J/(kg·K)) of material of the drum brake, while C_h is the specific heat capacity (J/(kg·K)) of the heating metal materials linking the drum brake; v_a is the initial velocity (m/s) of the brake drum; β is the braking force distribution coefficient. For the convenience of calculation, the brake mass m_d and the total mass of the metal materials linking the drum brake m_h are selected through the following formula:

$$\begin{aligned} m_d &= 2\rho_d e \cdot 2\pi R b_2, \\ m_h &= 2\rho_c \Delta \cdot 2\theta b_1 R, \end{aligned} \quad (2.7)$$

where ρ_d and ρ_c are the density of the brake drum and the density of the brake shoe, respectively. Let $\rho_d = \rho_c = 7570 \text{ kg/m}^3$; e and Δ are the thickness of brake drum and thickness of brake shoe, respectively, where $\Delta = 10 \text{ mm}$.

2.3. Constraint Conditions

(1) Self-Locking Constraint

The possibility of the self-locking for the brake shoe should be taken into consideration when designing the drum brake. To avoid self-locking, the following constraint should be satisfied:

$$g_1(X) = \mu - \frac{c' \cos \delta}{R_f - c' \sin \delta} < 0. \quad (2.8)$$

(2) Constraint of the Maximum Pressure on the Liners

The maximum pressure on the frictional liners should be less than the defined pressure value:

$$g_2(X) = \frac{\pi d^2 p (a + c) R_f}{b_1 R^2 [\cos \theta_0 - \cos(\theta_0 + \theta)] \cdot [c' (\cos \delta + \mu \sin \delta) - \mu R_f]} - 1.6 \leq 0. \quad (2.9)$$

(3) Constraint of the Uniformity of Pressure Distribution

The nonuniform distribution of pressure would cause uneven wear of the frictional liners, which would change the contact area, location, and pressure characteristic of the friction interface. Then the braking efficiency and stability is greatly lowered. The constraint is as follows:

$$g_3(X) = \frac{c' \theta}{(a + c) \sin(\theta/2)} - 2 \leq 0. \quad (2.10)$$

(4) Constraint of the Wearing Characteristic of Frictional Liners

During emergency braking, the specific energy dissipating rate of the frictional liners should be less than the defined rate:

$$g_4(X) = \frac{m_a v_a^2}{8t\theta R b_1} - E_u \leq 0, \quad (2.11)$$

where E_u is the upper limit of the specific energy dissipating rate of rear wheels, while the general upper limit of the specific energy dissipating rate of the brake drum is 1.9 W/mm^2 ; m_a is the total mass of the vehicle; R and b_1 are measured in millimeter and braking time $t = v_a/j$; v_a is the initial velocity (m/s); j is the braking deceleration (m/s^2), and in general $j = 0.6g$, where g is the gravity acceleration.

(5) Constraints of the Friction Area of Brake Lining

While braking, the frictional characteristic would be better if the positive pressure and energy load of the brake shoe lining area get lower. In cars, the general value range of the friction area of drum brake lining should obey the following constraints:

$$\begin{aligned} g_5(X) &= 10^4 - A_p \leq 0, \\ g_6(X) &= A_p - 3 \times 10^4 \leq 0, \end{aligned} \quad (2.12)$$

where $A_p = b_1 R \theta$ (mm^2).

(6) Heating Load Constraint

To calculate the heating load of the brake, the temperature rise of brake from the initial speed of 30 km/h to full stop should be less than the defined value $[\Delta t] = 15 \text{ K}$ [15]:

$$g_7(X) = \Delta t - [\Delta t] \leq 0. \quad (2.13)$$

(7) Constraints of the Radius of Brake Drum

Since the diameter of car brake drum is usually 64%~74% of rim diameter [10], thus, the ratio of car brake drum radius R against rim radius r_d should be in the range:

$$\begin{aligned} g_8(X) &= 0.64 - \frac{R}{r_d} \leq 0, \\ g_9(X) &= \frac{R}{r_d} - 0.74 \leq 0, \end{aligned} \quad (2.14)$$

where r_d is the rim radius (mm).

(8) Constraints of the Liner Width

When choosing the liner width, the pressure in a unit area should not be oversized since too large width would cause nonuniform contact. In general, the ratio of liner width against brake drum radius should meet the following constrains:

$$\begin{aligned} g_{10}(X) &= 0.32 - \frac{b_1}{R} \leq 0, \\ g_{11}(X) &= \frac{b_1}{R} - 0.54 \leq 0. \end{aligned} \quad (2.15)$$

3. Differential Cellular Multiobjective Genetic Algorithm

In order to solve the above-mentioned multiobjective optimization model much more efficiently, we introduce differential evolution (DE) strategy into the cellular genetic algorithm and thus form a new algorithm called differential cellular multiobjective genetic algorithm (DECell), which is able to minimize the distance from gained Pareto front set to exact optimal front while maintaining good uniformity and distribution range.

3.1. Differential Evolution Strategy

DE strategy is well suited for solving multiobjective problems of multidimension and non-linear as the main feature characterizing it is performing global parallel direct search by using the information of the distance and direction among the individuals in the population, and what is more, it can be easily implemented [16, 17]. The detailed operation process states as follows.

As for each individual $\vec{x}_{i,D}$, $i \in \{0, 1, \dots, N-1\}$, $i \in Z$, N means the population size, and the definition of the operation is:

$$\vec{v}_{i,D} = \vec{x}_{r1,D} + F \cdot (\vec{x}_{r2,D} - \vec{x}_{r3,D}), \quad (3.1)$$

where D is the number of the decision variables; $\vec{x}_{r1,D}$, $\vec{x}_{r2,D}$, and $\vec{x}_{r3,D}$ are the three selected parents; $r1$, $r2$, and $r3$ are the index positions of the parents; and F means the scaling factor. The previous formula can be explained as follows: select two parent vectors $\vec{x}_{r2,D}$ and $\vec{x}_{r3,D}$ randomly; get the difference value of the two vectors; add the zoomed difference to the third parent vector $\vec{x}_{r1,D}$, and then the individual vector after crossover is $\vec{v}_{i,D} = [v_{i,1}, \dots, v_{i,j}, \dots, v_{i,D}]$; as for each component $x_{i,j}$, which is in the parent vector $\vec{x}_{i,D} = [x_{i,1}, \dots, x_{i,j}, \dots, x_{i,D}]$ locating in the i th index position, generate randomly a number $\text{rand}_j \in [0, 1]$; make a comparison between rand_j and the crossover factor CR or between j and K to judge whether to replace $x_{i,j}$ with $v_{i,j}$; then get the new individual $\vec{u}_{i,D} = [u_{i,1}, \dots, u_{i,j}, \dots, u_{i,D}]$, where $K \in \{0, 1, \dots, D-1\}$, $K \in Z$:

$$u_{i,j} = \begin{cases} v_{i,j}, & \text{if } \text{rand}_j < \text{CR} \text{ or } j = K, \\ x_{i,j}, & \text{otherwise.} \end{cases} \quad (3.2)$$

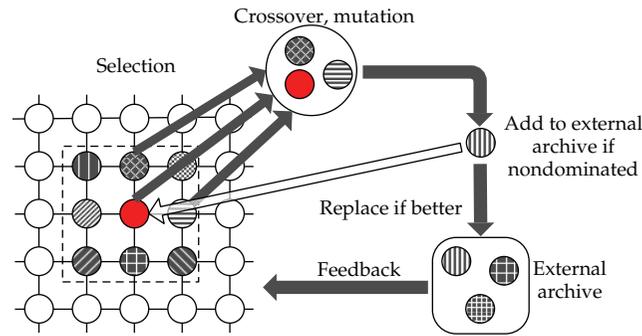


Figure 2: Breeding loop in DECell.

3.2. DECell Algorithm Principle

Cellular genetic algorithms (cGAs) are a kind of decentralized genetic algorithms (GAs) in which the population is structured in such way that the interaction of the individuals composing it is limited to a certain subset of individuals in the immediate vicinity of their location. Usually, the population is arranged in a 2D toroidal mesh in cGAs, and individuals are allowed to interact only with their nearby neighbors (see Figure 2 in the dotted box). This endows the cGA with useful properties for the optimization and also facilitates a parallel implementation because of its inherent parallel design [18].

Based on cGA, the detailed principle of DECell is presented in Figure 2. First, the population is structured in a two-dimension annular gird. Since cGAs are based on the concept of neighborhood, each individual (namely, current individual) has a set of close individuals or neighbors according to some regular structure (like Moore structure, von Neumann structure). In this paper, the neighborhood structure uses Moore structure, where current individual has eight nearby neighbors (see Figure 2). We select two better ones from the neighbors of current individual based on fitness. The three individuals containing the two selected individuals and current one compose the parent individuals, on which crossover operation is then performed by using DE strategy. An offspring is generated after mutation operation. If the offspring dominates the current individual, or both the offspring and the current individual are nondominated, but the offspring is with larger crowding distance than the current individual, then it will replace the current individual. Exploration and diffusion of solutions to the rest of the population happen in a smooth way, due to the continuous overlap existing among the different neighborhoods, which make high-quality solutions propagate over the population.

In the genetic operation process for each cellular individual, the nondominated individuals are stored in an external archive at the same time, and all the individuals in an external archive are ranked according to their crowding distance. If the number of the superior individuals exceeds the specified capacity of the archive, the individuals with smaller crowding distance will be removed. At the end of each generation, some of the individuals in the external archive are chosen randomly to replace the individuals of the same number in the original population so as to update the population constantly. By this method, the nondominated individuals in the external archive are able to obtain a reasonably good approximation to the

```

proc stepsUp()
  population←creatPopulation(); //Create initial population.
  archive←creatFront(); //Create a external archive for storing non-dominated solutions.
  while! StopCondition() //Termination condition.
    for individuals←1 to population.Size
      //Select the neighbors for the current individual (MOORE is adopted).
      neighborhood←selection(neighborhood);
      //Select two parent individuals from the neighbors.
      parent1←selection(neighborhood);
      parent2←selection(neighborhood);
      //Make sure the two parent individuals are different.
      while parent1==parent2
        parent2←selection(neighborhood);
      end while
      //Crossover operator with DE strategy and Pseudocode 2 presents
      the detailed operation.
      offspring←crossoverInDE(position(individual),position(parent1),
      position(parent2));
      offspring←Mutation(offspring); // Mutation.
      evaluationFitness(offspring); // Evaluate the offspring individuals.
      // The superior offspring replaces the present individual.
      insert(position(individual), offspring, population);
      // Put the non-dominated individuals into the external archive.
      addToArchive(individual);
    end for
    population←replaceIndividuals(population, archive); //Feedback mechanism.
  end while
end proc stepsUp;

```

PSEUDOCODE 1: DECell pseudocode.

Pareto optimal front when maintaining the diversity. The main pseudocode of DECell is in Pseudocode 1.

4. DECell Performance Test and Results Analysis

To assess how competitive the DECell is, a set of unconstraint, multivariable, and multiobjective test functions (i.e., DTLZ family) [19] and two constraint multiobjective test functions (i.e., the Viennet2 and Tanka) [20, 21] are adopted. DECell, NSGAI [22], MOCCell [23], PESA2 [24], and PAES [25] are used simultaneously to compute the above-mentioned test functions, respectively. The performances of the five algorithms are then compared and analyzed under the same assessment system.

4.1. Performance Assessment Indicators of Algorithms

This paper uses three assessment indicators: generational distance [26], spread [23], and hypervolume [27].

(1) *Generational Distance*

Generational distance is used to measure the distance between the obtained Pareto front and the Pareto optimal front. The computational formula is as follows:

$$GD = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{n}, \quad (4.1)$$

where n is the number of the obtained Pareto front and d_i is the Euclidean distance between the i th solution and the nearest solution in the Pareto optimal front in the objective space. It is obvious that all the obtained solutions are exactly the Pareto optimal solutions if $GD = 0$. Therefore, the smaller the value of GD is, the better it converges to the Pareto optimal solutions.

(2) *Spread*

The spread measurement proposed by Deb is an indicator that measures the distribution and spread of the obtained Pareto front. The calculation formula is as follows:

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{n-1} |d_i - \bar{d}|}{d_f + d_l + (n-1) \times \bar{d}}, \quad (4.2)$$

where d_i is the Euclidean distance between every two adjacent individuals in the obtained Pareto front, \bar{d} is the mean of these distances, d_f and d_l are the Euclidean distances between the boundary individuals of the obtained Pareto front and the boundary individuals of the Pareto optimal front, and n is the number of the individuals in the obtained Pareto front. A zero value of this indicator means an ideal uniform distribution. Therefore, the smaller this indicator's value is, the more uniform it distributes.

(3) *Hypervolume*

Hypervolume is adopted to calculate the volume covered by the individuals of the obtained Pareto front. It is defined as

$$HV = \text{volume} \left(\bigcup_{i=1}^{|Q|} v_i \right), \quad (4.3)$$

where Q is the amount of the obtained Pareto front; as for each individual i in the Pareto front, v_i is the hypercube formed by the reference point $W = (0, \dots, 0)$ and the individual i . Larger values of HV indicate that the obtained Pareto front individuals can be broadly distributed in the objective space.

4.2. Computational Results and Analysis

The parameters settings of DECell, NSGA-II, MOCcell, PESA2, and PAES are as follows: all the algorithms adopt real coding and polynomial mutation. NSGAI, MOCcell, PESA2, and PAES employ SBX [28] crossover operator, while DECell adopts DE crossover operator. Let $F = 0.1$

Table 1: Average (avg) and standard deviation (std) of the GD indicator of the Pareto front obtained by five algorithms for different problems.

MOP	DECell		MOCcell		NSGA-II		PESA2		PAES	
	avg	std								
DTLZ1	5.30e-2	5.92e-2	1.47e-1	1.38e-1	1.49e-1	2.76e-1	3.47e-1	4.45e-1	7.94e-1	7.87e-1
DTLZ2	1.38e-3	1.22e-4	1.65e-3	5.55e-4	1.43e-3	2.13e-4	1.39e-3	7.95e-5	1.41e-3	2.51e-4
DTLZ3	4.73e-1	3.26e-1	2.25	1.32	2.75	1.33	2.11	1.53	4.10	2.95
DTLZ4	1.09e-2	1.62e-2	3.76e-3	2.18e-3	5.17e-3	3.11e-4	2.88e-3	2.13e-3	5.31e-4	1.29e-3
DTLZ5	7.07e-4	1.30e-4	2.64e-4	2.07e-5	3.87e-4	5.96e-5	5.65e-4	1.14e-4	4.12e-4	1.30e-4
DTLZ6	8.17e-4	1.24e-4	4.26e-1	4.10e-2	2.48e-1	2.06e-2	1.08e-1	7.94e-3	3.61e-2	3.24e-2
Viennet2	8.32e-4	2.36e-4	8.32e-4	2.45e-4	9.10e-4	2.82e-4	8.33e-4	4.12e-4	1.21e-3	5.24e-4
Tanka	3.13e-4	7.85e-5	6.41e-4	8.71e-5	7.34e-4	8.29e-5	8.91e-4	1.30e-4	7.36e-4	1.33e-4

Table 2: Average (avg) and standard deviation (std) of the spread (Δ) indicator of the Pareto front obtained by five algorithms for different problems.

MOP	DECell		MOCcell		NSGA-II		PESA2		PAES	
	avg	std								
DTLZ1	6.27e-1	3.58e-2	9.03e-1	1.36e-1	9.70e-1	1.53e-1	9.35e-1	2.32e-1	1.19	2.75e-1
DTLZ2	5.72e-1	1.22e-4	7.20e-1	4.11e-2	6.92e-1	5.51e-2	6.95e-1	4.41e-2	6.59e-1	6.50e-2
DTLZ3	6.31e-1	8.72e-2	1.09	8.31e-2	1.09	1.25e-1	1.07	1.19e-1	1.4	1.30e-1
DTLZ4	7.85e-1	2.37e-1	7.63e-1	1.45e-1	6.61e-1	4.52e-2	7.89e-1	1.65e-1	1.02	9.15e-2
DTLZ5	4.44e-1	7.05e-2	1.50e-1	3.62e-2	4.50e-1	3.34e-2	6.69e-1	3.34e-2	6.70e-1	5.66e-2
DTLZ6	7.51e-1	1.08e-1	7.55e-1	3.69e-2	7.97e-1	4.03e-2	6.82e-1	4.88e-2	1.04	1.87e-1
Viennet2	7.25e-1	4.06e-2	8.53e-1	7.01e-2	8.49e-1	8.01e-2	1.01	5.03e-2	1.03	5.49e-2
Tanka	2.90e-1	3.95e-2	7.40e-1	3.21e-2	8.17e-1	2.57e-2	1.15	3.47e-2	1.28	6.76e-2

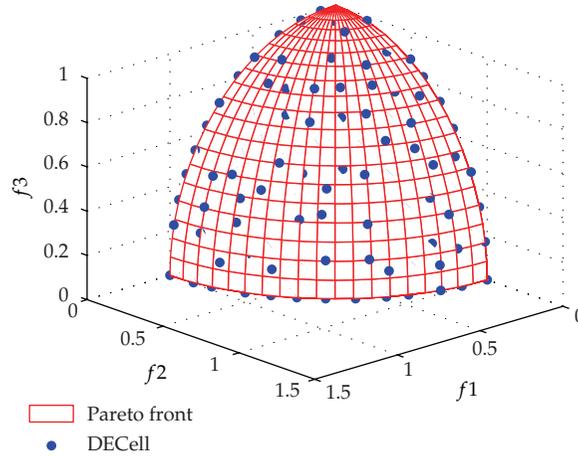
and CR = 0.5. The population size is 100, the maximum evaluation generation is 20000, and crossover probability is 0.9, while mutation probability is $1/\text{len}$, where “len” means the variable length. These five algorithms run 20 times independently for every test function, respectively. Quantitative analysis on the performances of these five algorithms is presented in Tables 1, 2, and 3, in which the values in the dark grey blanks are the optimal values, while those in light grey blanks are the second optimal values.

We consider first the generational distance (GD). As the test results of the computation on the benchmarks in Table 1 show that, DECell can get the best indicator values in six of the eight multiobjective problems, MOCcell and PESA2 can only obtain the best values in one of the eight test functions, respectively. Moreover, NSGA-II and PAES cannot get any best GD value in any of the problems. Therefore, a conclusion can be made that in general DECell outperforms the other four algorithms in terms of convergence when solving high-dimensional multiobjective optimization problems (MOPs) mentioned previously.

Regarding the spread value (Table 2), the results reveal that DECell outperforms the other four algorithms in terms of the diversity of the nondominated solutions along with the obtained Pareto front since it obtains the best (lowest) spread value in five of the eight multiobjective functions, along with two suboptimal values. The Δ indicators suggest that PAES performs the worst, as it cannot get best value in any problem, only with one suboptimal value.

Table 3: Average (avg) and standard deviation (std) of the HV indicator of the Pareto front obtained by five algorithms for different problems.

MOP	DECell		MOCcell		NSGA-II		PESA2		PAES	
	avg	std								
DTLZ1	$4.13e-1$	$3.68e-1$	$3.68e-1$	$3.44e-1$	$5.49e-1$	$2.07e-1$	$3.94e-1$	$2.96e-1$	$4.73e-1$	$9.26e-2$
DTLZ2	$4.06e-1$	$1.64e-2$	$3.74e-1$	$4.40e-3$	$3.73e-1$	$5.51e-3$	$3.30e-2$	$2.36e-2$	$2.24e-1$	$6.67e-2$
DTLZ3	$2.14e-2$	$8.69e-2$	0		0		0		$1.24e-2$	$2.53e-2$
DTLZ4	$3.20e-1$	$1.23e-1$	$2.85e-1$	$1.65e-1$	$3.84e-1$	$5.74e-3$	$2.46e-2$	$1.15e-1$	$2.02e-2$	$5.68e-2$
DTLZ5	$9.30e-2$	$2.95e-4$	$9.27e-2$	$2.98e-5$	$9.25e-2$	$2.28e-4$	$9.12e-2$	$3.01e-4$	$9.18e-2$	$3.96e-4$
DTLZ6	$8.79e-2$	$3.51e-3$	0		0		0		$8.17e-2$	$2.30e-2$
Viennet2	$9.26e-1$	$3.85e-4$	$9.22e-1$	$9.69e-4$	$9.21e-1$	$1.57e-3$	$9.13e-1$	$4.47e-3$	$9.12e-1$	$4.03e-3$
Tanka	$5.38e-1$	$8.18e-4$	$3.09e-1$	$3.85e-4$	$3.07e-1$	$4.30e-4$	$3.03e-1$	$7.45e-4$	$3.01e-1$	$2.05e-3$

**Figure 3:** DTLZ2 solved by DECell.

The values of hypervolume indicator (HV) in Table 3 indicate that DECell gets better results than the other algorithms. DECell obtains the best values (the highest values) in six of the eight test functions, while NSGA-II gets the best values in two of the eight functions, which denotes that the Pareto front individuals obtained by DECell are able to be distributed more broadly in the objective space.

To compare the performance of algorithms more clearly, we present Figures 3, 4, 5, 6, and 7 the Pareto fronts gained by DECell, MOCcell, NSGA-II, PESA2, and PAES when solving DTLZ2, where the grid areas are the Pareto optimal front surface and the dots represent the Pareto front obtained by each algorithm. It is clear that DECell outperforms the other four algorithms concerning distribution and convergence when solving DTLZ2. The PESA2 has the worst distribution, while MOCcell has the worst convergence. And as for the coverage of the gained solutions, DECell outperforms the other ones, which is followed by MOCcell and NSGA-II, and next is PAES, while PESA2 is the worst, which matches the results shown in Tables 1, 2, and 3.

Another analysis method is needed to further analyze the influence of the evaluation number on the algorithm performance. The simple way is to observe the evolution of three given indicators during the whole execution of the algorithms from which we can tell what

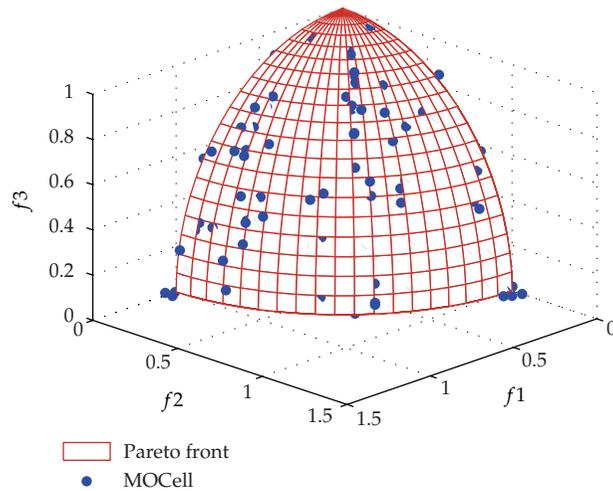


Figure 4: DTLZ2 solved by MOCell.

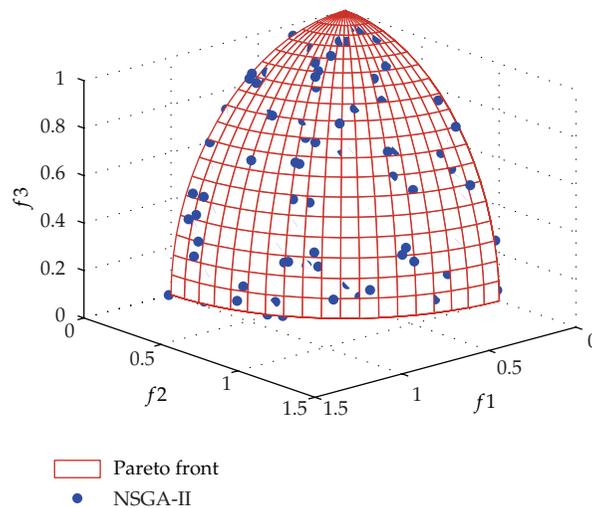


Figure 5: DTLZ2 solved by NSGA-II.

is happening in every generation. Figure 8 shows the convergence progress of GD, HV, and spread when solving DTLZ1 problem with different evaluation number. In this figure, with evaluation number increasing, we can see that the GD values obtained by all the algorithms have converged to the lowest (or best) value and the HV value obtained by DECell is larger (or better) than those obtained by the other algorithms during evolution. We also observe that fluctuation in spread value is unstable, but the fluctuation of DECell is smaller than that of the other algorithms, and the spread value obtained by DECell is lower (or better) than those by the other algorithms with the progress of evolution as well.

A more detailed analysis can be performed if we show the results using boxplots, which constitutes a useful way of depicting groups of numerical data. Figure 9 presents the boxplot distribution of GD, spread, and HV values obtained by the five algorithms in ZTLZ3.

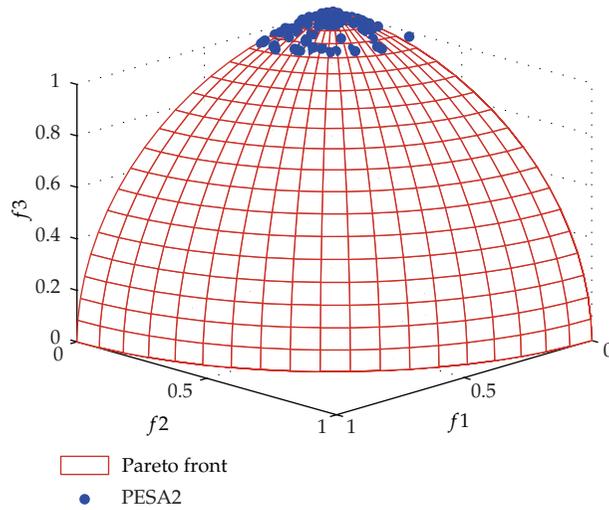


Figure 6: DTLZ2 solved by PESA2.

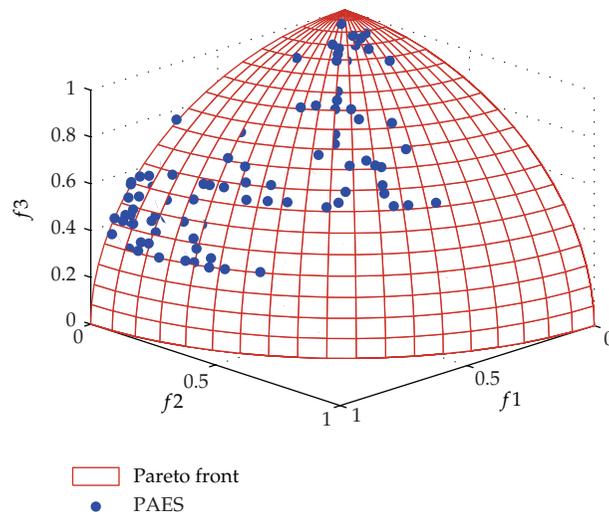


Figure 7: DTLZ2 solved by PAES.

We can see that GD and spread values obtained by DECell have been lower (or better) than the ones obtained by the other algorithms though HV values obtained by these algorithms vary slightly (in fact, HV values have no confidence interval).

It can be concluded that in general DECell performs better than the other algorithms when solving complicated multiobjective problems, while NSGA-II takes the second place. Though DECell is outstanding in accuracy, it is worse than some other MOEAs in efficiency. That is to say, DECell costs more computational time than NSGA-II, PESA2 for the problems, because DECell is a kind of decentralized cellular genetic algorithms, where individual interacts only with its nearby neighbors and diffusion of solutions to the population happens slowly in a smooth fashion.

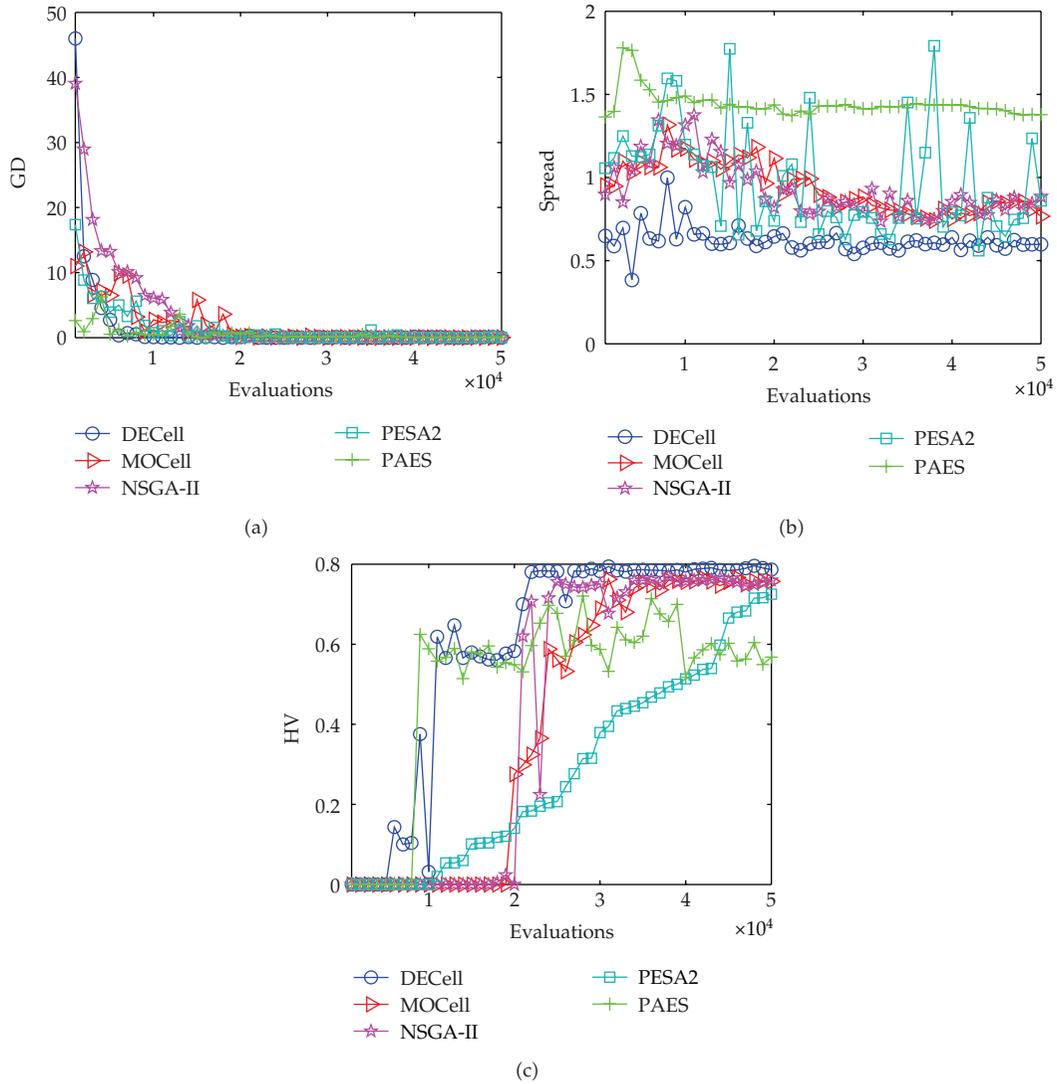


Figure 8: Evolution of the three indicators for different algorithms in ZDT1.

5. Parameter Optimization Design of Brake Based on DECell

The computational procedures of the brake model by using DECell are as follows.

(1) Population Initialization

Randomly generate initial population in the constraint range of optimization parameters by real coding. The optimization parameters are $X^T = [x_1, x_2, \dots, x_{11}] = [\theta_0, \theta, b_1, R, c', c, a, b_2, e, d, p]$.

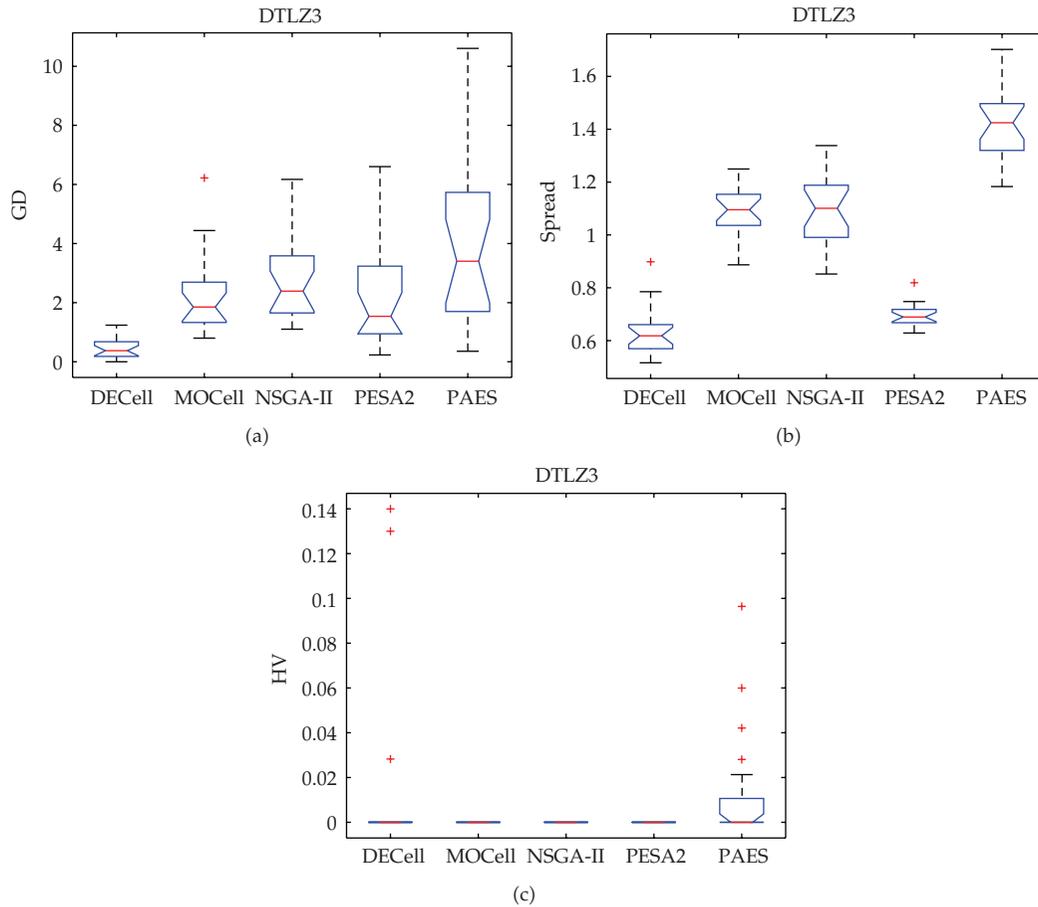


Figure 9: Boxplot representation of GD, spread, and HV values of the fronts computed by five algorithms in ZTLZ3.

(2) Selection

The tournament method is used, which selects the parent individuals based on the individual rank and crowding distance. The detailed procedures are as follows: select the neighbors of the current individual; compare the ranks of the neighbors, and remain the individuals with lower ranks since the individuals with lower ranks performs better; if the neighbors are of the same rank, compare their crowding distances, and reserve the ones with bigger crowding distances. At last, two better individuals are selected from the neighbors.

(3) Crossover

Crossover operation is performed to generate more new individuals and improve the space searching ability. DE strategy is introduced to the crossover operation, and the detailed procedures are explained in formulas (3.1) and (3.2). The parameters CR and F are adopted in this operation, where the crossover coefficient (CR) equals 0.5 and the scaling factor (F)

```

proc crossoverInDE(r1, r2, r3)
//  $P_{x_1}$  and  $P_{x_2}$  are the corresponding initial position index and terminal position
index of the parameters
Randj = randInt( $P_{x_1}, P_{x_2}$ );
// perform differential crossover operation
for ( $j = P_{x_1}, j \Leftarrow P_{x_2}, j = j + 1$ )
    if (rand(0,1) < CR or  $j = \text{Rand}_j$ )
         $v_{r3[j],t} = x_{r3[j],t} + F \times (x_{r1[j],t} - x_{r2[j],t})$ ; //  $t$  means current functions evaluations
    else
         $v_{r3[j],t} = x_{r3[j],t}$ ;
    end if
end for
return  $v_{r3,t}$ ;
end_proc crossoverInDE

```

PSEUDOCODE 2: Crossover operation pseudocode based on DE strategy.

Table 4: Model parameters.

Parameters	The weight of the vehicle m_a (kg)	Distribution coefficient β	Rim radius R_r (mm)	Initial velocity of brake v_a (km/h)	Friction coefficient of the friction plate μ	Specific heat capacity C_d, C_h (J/(kg·K))
Values	1560	0.77	177.8	30	0.35	482

equals 0.5. The detailed parameter setting is as per [29]. The code of the crossover flow is presented in Pseudocode 2.

(4) Mutation

This operation aims to avoid local convergence.

(5) Termination Condition

If the maximum evaluation generations are reached, output the results or go back to step (2).

6. Case Study

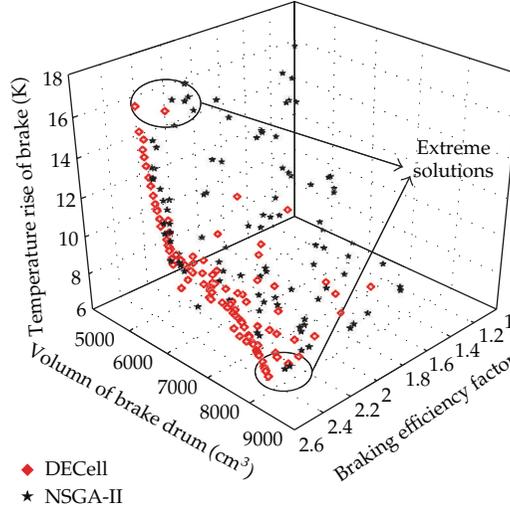
For a vehicle, some of its parameters and its leading- and trailing-shoe brake are shown in Table 4.

To search optimal solutions of the multiobjective problems, Table 5 presents the boundary of design variables.

To evaluate the competitiveness of this algorithm, the most commonly used multiobjective genetic algorithm NSGA-II is also adopted to compute the brake model besides DECell. The parameters settings are as follows: population size $M = 100$, archive capacity of DECell $N = 100$, feedback number $C = 25$, crossover probability $p_c = 0.9$, and mutation probability $p_m = 1/\text{var}$, where “var” is the number of the decision variables.

Table 5: Domain of decision variables.

Variable	a' (degree)	β (degree)	b_1 (mm)	R (mm)	c' (mm)	c (mm)
Upper limit	10	90	36	120	120	85
Lower limit	45	120	75	180	160	120
Variable	a (mm)	b_2 (mm)	e (mm)	d (mm)	p (MP)	
Upper limit	60	90	5	15	6	
Lower limit	130	120	16	35	12	

**Figure 10:** Distribution of the Pareto front.

6.1. Qualitative Analysis

After each evolution, the superior solutions gained by NSGA-II are put in the final population, while those obtained by DECell are stored in the external archive. Since the population size and the archive capacity are both 100, each algorithm obtains 100 superior solutions after computation. However, there are still some dominated solutions existing in the 100 individuals, which means that there would be 100 or less non-dominated ones after each computation.

Let the maximum evaluation generations be $G = 25000$. Two algorithms are operated under the same conditions; they both get 100 nondominated individuals, which are shown in Figure 8.

Even though we cannot easily and specifically tell which algorithm performs better in terms of the three objectives from Figure 10, the overall situation shows that the braking efficiency factor obtained by DECell is obviously bigger than that got by NSGA-II. Since the bigger the braking efficiency factor is, the better the braking performance is, it is clear that DECell outperforms NSGA-II for the objective of braking efficiency. In order to make clear the relationship between temperature rise and the volume of the brake drum, the two-dimensional graph of the two factors is shown separately in Figure 11.

In Figure 11, the Pareto fronts in upper left are of higher temperature rise and smaller brake drum volume, which means that the smaller the brake drum is, the higher the brake temperature rise will be, that is, the heat dispersion is worse. As for the Pareto fronts in lower

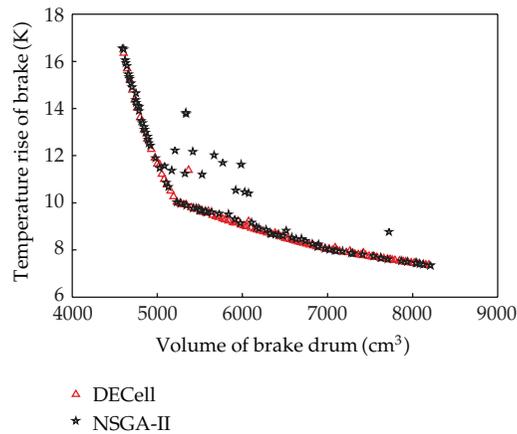


Figure 11: Pareto front set schematic diagram between volume and temperature rise.

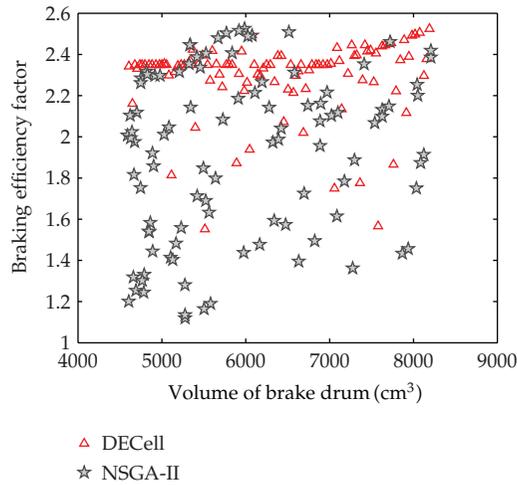


Figure 12: Pareto front set schematic diagram between volume and efficiency factor.

right, they have lower brake temperature rise and larger brake drum volume, which indicates that bigger brake drum volume accompanies smaller brake temperature rise; that is, the heat dispersion is much better. In Figure 11, as for the two factors, the Pareto front individuals gained by DECell distribute more uniform than those of NSGA-II; moreover, it is of better continuity. What is more, more Pareto front individuals obtained by DECell are distributed in the compromise position. It can be concluded that DECell outperforms NSGA-II in terms of distribution and convergence.

Figure 12 presents the relationship between braking efficiency factor and the volume of the brake drum. In the figure, Pareto front solutions obtained by the two algorithms distribute disorderly in 2D multiobjective space, which implies that the efficiency factor is irrelative with the volume of drum brake. That is to say, there is no promoting or conflicting relationship between them.

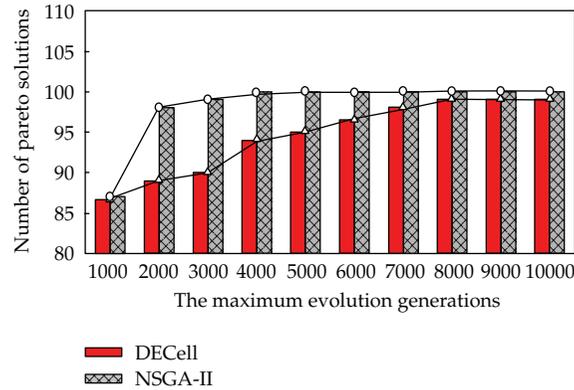


Figure 13: The number of the nondominated individuals obtained by the two algorithms.

6.2. Quantitative Analysis

6.2.1. Convergence Rate Analysis

To further assess the algorithms performance in searching Pareto optimal solutions, different maximum evolution generations, which are got from 1000 to 10000 with the interval 1000, are adopted. For every maximum evolution generation, the aforementioned two algorithms are applied to compute the model of brake for 20 times, respectively, and the averages of the number of the nondominated individuals gained by each algorithm during the 20 times computations are plotted in Figure 13.

Figure 13 shows that DECell can always find more nondominated individuals than NSGA-II with the same maximum evolution generations and the former algorithm outperforms the latter in terms of space searching ability, optimal solution saving ability, and exploring ability.

6.2.2. Convergence Accuracy Analysis

To further analyze the computational stability and accuracy of the algorithm, the maximum evolution generation is set as $G = 25000$ and 20 independent computations on the brake optimization design model are performed by NSGA-II and DECell, respectively. Each algorithm gets 2000 Pareto solutions (i.e., 2000 sets of objective function values are obtained), which are then statistically analyzed in terms of three indicators: the mean, standard deviation and optimal value of all the values gained for each objective. The results are presented in Table 6.

Each data in the grey blanks presented in Table 6 is the superior value of the two data in the same type. If the brake is of large braking efficiency factor, small volume, and low temperature rise, it is of good performance.

As is shown in Table 6, DECell gets better mean efficiency factor and temperature rise values, except that the average volume of drum brake optimized by DECell is a little larger than that of NSGA-II (the volume is still acceptable). The results denote that DECell outperforms NSGA-II in finding much better design parameters of brake.

Table 6: Performance comparison.

Algorithm comparison	NSGA-II			DECell		
	Braking efficiency factor	Drum brake volume	Temperature rise of brake	Braking efficiency factor	Drum brake volume	Temperature rise of brake
Mean	1.951	$6.14e + 3$	10.257	2.246	$6.46e + 3$	9.20
Standard deviation	0.041	70.864	0.120	0.079	43.645	0.102
Optimal value	2.525	$4.756e + 3$	7.433	2.526	$4.756e + 3$	7.431

Meanwhile, as for the deviation of three indicators, two of them obtained by DECell are lower than those obtained by NSGA-II, which means that DECell is of low volatility and good stability. As for each objective, all the optimal values found by DECell are better than those got by NSGA-II. It indicates that DECell can converge to the much better Pareto solutions and it is of higher accuracy.

Therefore, it can be concluded that differential cellular multiobjective genetic algorithm performs better than NSGA-II in terms of solution space searching ability and speed for finding much more and better solutions when solving the brake optimization design model. Moreover, the solutions could be kept effectively. What is more, this algorithm is of good stability and high accuracy, in which much more good design parameters of brake could be easily found for designers to choose.

7. Conclusions

This paper established a multiobjective optimization design model of drum brake with the goals of maximizing the efficiency factor of braking, minimizing the volume of drum brake, and minimizing the temperature rise of brake, in order to better meet the requirements of engineering practice.

We present a new optimization algorithm by introducing the differential strategy into the canonical cellular genetic algorithm to solve the model effectively, which is called differential evolution cellular multiobjective genetic algorithm (DECell). In DECell, all the population individuals are structured in a bidimensional grid, and each individual may only interact with its neighborhood. Besides, three parent individuals are selected for each individual to take part in the crossover operation together. With this method, the obtained Pareto front would be as approximate as possible to the Pareto optimal front. Besides, their distribution uniformity and coverage are well maintained. The evaluation on test function indicates that the proposed algorithm outperforms the other four popular multiobjective algorithms (i.e., NSGAI, MOCell, PESA2, and PAES) greatly when solving high-dimensional nonlinear multiobjective problems. The experimental computation of DECell and NSGA-II on parameter optimization design of drum brake suggests that the DECell can find much more and better solutions stably. It is an effective algorithm that can be applied to solve the drum brake parameters optimization and other complicated engineering problems.

Acknowledgment

This work was supported by the National Science Foundation of China (51275274).

References

- [1] L. Cheng and W. M. Zhang, "Optimization design of mining trucks & caliper disc brake," *Mining Research and Development*, vol. 28, no. 1, pp. 41–43, 2008.
- [2] J. Li, "Optimization design of tractor brake based on hybrid PSO," *Chinese Agricultural Mechanization*, vol. 4, pp. 93–96, 2011.
- [3] Z. H. Li, X. L. Zhang, and L. C. Guo, "Optimal design for caliper disc brake," *Machine Design and Research*, vol. 25, no. 2, pp. 83–85, 2009.
- [4] L. Chen, "Simulated annealing algorithm for function optimization and application in caliper-type brake," *Coal Mine Machinery*, vol. 27, no. 10, pp. 38–41, 2006.
- [5] Q. B. Huang, M. G. Zhou, and Y. Wang, "Structural parameters optimization design of drum brake based on genetic algorithm," *Machinery*, vol. 11, pp. 24–26, 2006.
- [6] P. Jiang and W. J. Li, "Optimal design of disc brake based on MATLAB," *Mechanical Engineering & Automation*, vol. 6, pp. 158–161, 2007.
- [7] J. Mi and Y. L. Wu, "Multi-objective optimization design of drum-fashioned brake technology," *Machinery Design & Manufacture*, vol. 1, pp. 25–26, 2007.
- [8] J. Mi and Y. L. Wu, "The Application of multi-objective optimization method in brake design," *Journal of Mechanical Transmission*, vol. 31, no. 5, pp. 71–73, 2007.
- [9] G. L. Qin, D. M. Wang, and X. H. Chen, "Optimization design of disk brake based on an improved genetic algorithm," *Modular Machine Tool & Automatic Manufacturing Technique*, vol. 7, pp. 101–103, 2011.
- [10] H. Wang, *Parametric Brake Optimization Based on Ant Colony Algorithm*, Jilin University, Jilin, China, 2007.
- [11] J. Wu and W. J. Li, "Optimal design of disk brake based on improved swarm optimization," *Machinery Design & Manufacture*, vol. 4, pp. 18–20, 2007.
- [12] R. H. Yang, "Multi-objective optimization design of auto drum-fashioned brake based on genetic algorithm of MATLAB," *Machine Tool & Hydraulics*, vol. 23, pp. 91–93, 2011.
- [13] M.G. Zhou, Y. Chen, J. D. Zhou, and J. Yi, "Genetic algorithm optimal design and its implementation with MATLAB for drum brake," *Journal of Hubei University of Technology*, vol. 24, no. 2, pp. 78–80, 2009.
- [14] W. Y. Wang, *Automotive Design*, China Machine Press, Beijing, China, 2004.
- [15] W. X. Liu, *Structural Analysis & Design of Vehicle Brake System*, Tsinghua University Press, Beijing, China, 2004.
- [16] Y. J. Shi, *The Cooperative Co-Evolutionary Differential Evolution and Its Applications for Complex Layout Optimization*, Dalian University of Technology, Dalian, China, 2006.
- [17] J. Durillo, A. J. Nebro, F. Luna, and E. Alba, "Solving three-objective optimization problems using a new hybrid cellular genetic algorithm," in *Proceedings of the 10th international conference on Parallel Problem Solving from Nature: PPSN X*, pp. 661–670, 2008.
- [18] B. Dorronsoro and E. Alba, "A simple cellular genetic algorithm for continuous optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'06)*, pp. 2838–2844, IEEE Computer Society, Vancouver, Canada, July 2006.
- [19] Deb, K. L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multi-objective optimization," Tech. Rep. 112, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, 2001.
- [20] A. Vicini and D. Quagliarella, "Multipoint transonic airfoil design by means of a multiobjective genetic algorithm," Tech. Rep. AIAA-97-0082, AIAA, Washington, DC, USA, 1997.
- [21] H. Tamaki, M. Mori, M. Araki, Y. Mishima, and H. Ogai, "Multi-criteria optimization by genetic algorithms: a case of scheduling in hot rolling process," in *Proceedings of the 3rd Conference of the Association of Asian-Pacific Operational Research Societies with IFORS (APORS'94)*, pp. 374–381, World Scientific, Fukuoka, Japan, July 1994.
- [22] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [23] A. J. Nebro, J. J. Durillo, F. Luna, B. Dorronsoro, and E. Alba, "MOCcell: a cellular genetic algorithm for multiobjective optimization," *International Journal of Intelligent Systems*, vol. 24, no. 7, pp. 726–746, 2009.
- [24] Corne, D. W. N. R. Jerram, J. D. Knowles, M. J. Oates, and J. Martin, "PESA-II: Region-based selection in evolutionary multi-objective optimization," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'01)*, pp. 283–290, Morgan Kaufmann, San Francisco, Calif, USA, July 2001.

- [25] J. D. Knowles and D. W. Corne, "Approximating the nondominated front using the Pareto archived evolution strategy," *Evolutionary Computation*, vol. 8, no. 2, pp. 149–172, 2000.
- [26] D. A. van Veldhuizen and G. B. Lamont, "On measuring multiobjective evolutionary algorithm performance," in *Proceedings of the Congress on Evolutionary Computation (CECCO'00)*, pp. 204–211, Morgan Kaufmann, Las Vegas, Nevada, USA, 2000.
- [27] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.
- [28] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 2, pp. 115–148, 1995.
- [29] K. Zielinski, P. Weitkemper, R. Laur, and K. D. Kammeyer, "Parameter study for differential evolution using a power allocation problem including interference cancellation," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'06)*, pp. 1857–1864, IEEE Computer Society, Vancouver, Canada, July 2006.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

